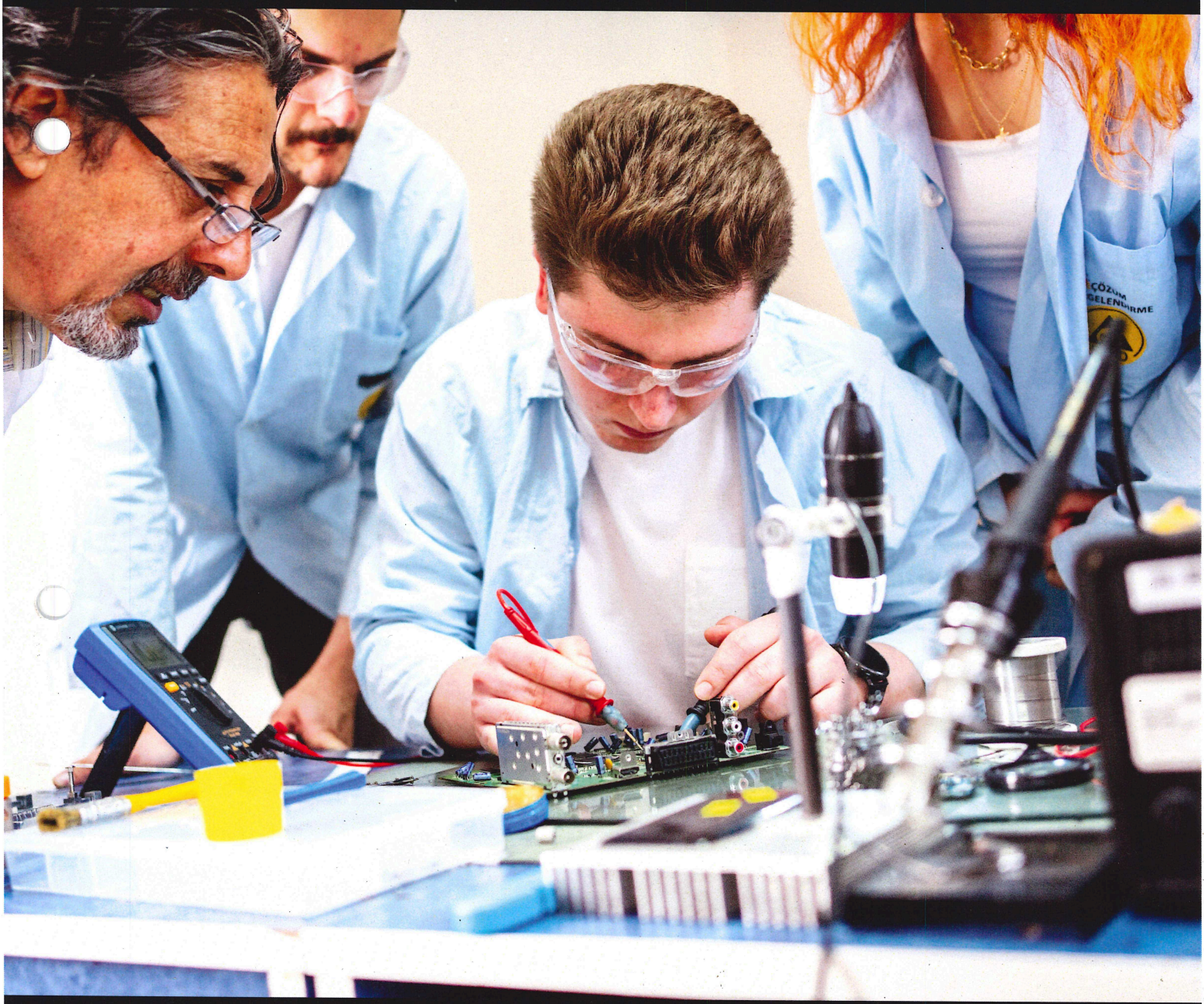
TAFE NSW

Digital Electronics - Part 2



tafensw.edu.au

Workbook

Author: Barry Duncan
This workbook is copyright protected and must not be reproduced without written consent (except under the *fair dealing* exceptions specified in the copyright act). For more information on what constitutes fair dealing visit: www.copyright.org.au

---

Digital Electronics - Part 2

Version 2.0 - January 2025

Printed for exclusive use by students of:
The Electronics Section
Ultimo Campus of TAFE NSW
Mary Anne St
Ultimo NSW 2007

Contents

Introduction _____ I-3

Textbooks _____ I-4

Lesson organiser _____ I-5

Section 1 The D-type flip flop _____ 1-1

Skill Practice 1 _____ 1-19

Review questions _____ 1-25

Section 1 The J-K flip flop _____ 2-1

Skill Practice 2 _____ 2-13

Review questions _____ 2-21

Section 3 Multivibrators _____ 3-1

Skill Practice 3 _____ 3-11

Review questions _____ 3-17

Section 4 Counters _____ 4-1

Skill Practice 4 _____ 4-19

Review questions _____ 4-29

Section 5 Shift registers _____ 5-1

Skill Practice 5 _____ 5-25

Review questions _____ 5-33

Section 6 Encoders and decoders _____ 6-1

Skill Practice 6 _____ 6-19

Review questions _____ 6-29

Section 7 Multiplexers and demultiplexers _____ 7-1

Skill Practice 7 _____ 7-19

Review questions _____ 7-29

Section 8 Analog to digital and digital to analog converters _____ 8-1

Skill Practice 8 _____ 8-19

Review questions _____ 8-33

Appendices

Appendix 1: Flip flop circuits that implement counters

Appendix 2: Circuits that implement encoders and decoders

Appendix 3: Circuits that implement multiplexers and demultiplexers

Appendix 4: ADC structures

Appendix 5: DAC circuits

Appendix 6: Digital IC data

Answers

Section 1 _____ A-1

Section 2 _____ A-4

Section 3 _____ A-8

Section 4 _____ A-12

Section 5 _____ A-15

Section 6 _____ A-16

Section 7 _____ A-18

Section 8 _____ A-20

Introduction

This workbook is the second of two that have been written to help students satisfy the assessment requirements for the following Units:

- UEEEC0069 - Troubleshoot digital sub-systems

- More information about electrotechnology training packages and the above Units can be found at www.training.gov.au/home/tga. Type the course/unit number in the "Quick search" and press return.

Textbooks

The following textbooks can be used to supplement the notes in this workbook. Most (if not all) of these books are available from the library in Building D.

- Bignell (1994). *Digital electronics* (3rd edition). Delmar Publishers.
- Bogart (1992). *Introduction to digital circuits*. McGraw Hill.
- Cook (1999). *A first course in digital electronics*. Prentice Hall.
- Edwards & Myer (1993). *Electronics; a basic course* (2nd edition). McGraw Hill.
- Floyd (2002). *Digital fundamentals* (7th edition). Prentice Hall.
- Gates (2001). *Introduction to electronics* (4th edition). Thompson Printing.
- Green (1993). *Digital electronics for technicians* (4th edition). Longman Scientific and Technical.
- Kleitz* (2002). *Digital electronics: A practical approach* (6th edition). Prentice Hall.
- Maxfield (2003). *Bebop to the Boolean boogie: an unconventional guide to electronics fundamentals, components and processes* (2nd edition). Newnes.
- Predko (2005). *Digital electronics demystified*. McGraw Hill.
- Tocci, Moss & Widmer (2004). *Digital systems: Principles and applications* (9th edition). Prentice Hall.
- Tokheim (2003). *Digital electronics: Principles and applications* (6th edition). McGraw Hill.
- Thompson (2001). *Digital electronics: A simplified approach*. Prentice Hall.

* Recommended

Lesson organiser

The following is a suggestion only. It may be changed to meet student needs and/or different modes of delivery.

Lesson	Section	Activity
10	Pt 2 - 1	The D-type flip flop Skill practice 1 - Practise testing and measuring sequential logic circuits (D-type flip flops)
11	Pt 2 - 2 Pt 2 - 3	The J-K flip flop Skill practice 2 - Practise testing and measuring sequential logic circuits (J-K flip flops) Multivibrators Skill practice 3- Practise testing and measuring monostable multivibrator circuits
12	Pt 2 - 4	Counters Skill practice 4 - Practise testing and measuring sequential logic digital couner ICs
13	Pt 2 - 5	Shift registers Skill practice 5 - Practise testing and measuring sequential logic shift register ICs
14	Pt 2 - 6	Encoders and decoders Skill practice 6 - Practise testing and measuring digital decoder ICs
15	Pt 2 - 7	Multiplexers and demultiplexers Skill practice 7 - Practise testing and measuring digital multiplexer and demultiplexer ICs
16	Pt 2 - 8	Analog to digital and digital to analog converters Skill practice 8 - Practise testing and measuring ADCs and DACs
17	■ **Possible assessment - Refer to the UAG for details**	
18	■ **Possible assessment - Refer to the UAG for details**	

Copyright: Not to be reproduced without permission

Student notes

Section 1 The D-type flip flop

Purpose To develop your understanding of the D-type flip flop and to develop your ability to interpret manufacturers' data sheets.

Objectives At the end of this section you should be able to:

- Explain the difference between *level-sensitive* and *edge-sensitive* trigger inputs to sequential logic devices
- Explain the difference between *positive edge-triggered* and *negative edge-triggered* clock inputs
- Recognise the schematic symbol of a D-type flip flop
- Predict a D-type flip flop's output for a given set of inputs
- Draw and/or interpret a timing diagram for a D-type flip flop (with and without preset and clear inputs) given a set of logic levels for their inputs that change over time
- Explain the problem that can be caused when preset and clear inputs on logic devices (like D-type flip flops) are left floating and how it can be prevented
- Give applications for the D-type flip flop
- Interpret manufacturers' data for common commercially available D-type flip flops
- Use a logic probe to test and measure sequential logic circuits

Introduction

The last chapter of the Part 1 workbook explained the operation of combinational logic RS flip flops. This was followed by a discussion of the differences between combinational logic and sequential logic. Finally, triggering principles related to sequential logic devices with active high and active low gate/enable inputs were demonstrated using RS flip flops.

This first chapter of the Part 2 workbook introduces you to the D-type flip flop. Importantly, D-type flip flops have an edge-sensitive trigger input and so this issue is explained first.

Edge-sensitive trigger inputs

Although many devices have level sensitive trigger inputs (which are usually called *gate*, *enable* or *strobe*), they are impractical where devices are required to read the instantaneous value of any of their inputs. For these situations *edge-sensitive* trigger inputs are needed which only allow the synchronous inputs of the device to be read during the trigger signal's transition from a logic-0 to a logic-1 or vice versa. Such trigger inputs are usually called the *clock* (*CLK*).

A positive edge-sensitive clock input allows the device to read its synchronous inputs when the clock signal changes from a logic-0 to a logic-1 (see Figure 1a). A negative edge-sensitive clock input allows the device to read its synchronous inputs when the clock signal changes from a logic-0 to a logic-1 (see Figure 1b). Clock inputs are usually referred to as being positive edge-triggered and negative edge-triggered.

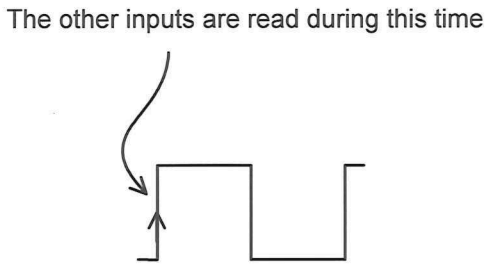


Figure 1a *Positive edge-triggered clock*

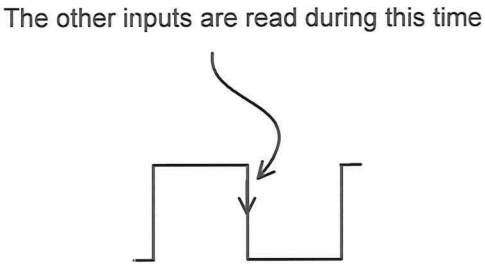


Figure 1b *Negative edge-triggered clock*

The D-type Flip Flop

Figure 2 below shows the schematic symbols of D-type flip flops with a positive edge-triggered clock input (left) and a negative edge-triggered clock input (right).

Figure 2 *The schematic symbols of D-type flip flops that are: a) positive edge-triggered; and b) negative edge-triggered*

Like the RS FF, the D-type FF has a Q and $\overline{Q}$ output with logic levels that are always the opposite logic level to each other. The *CLK* input is an edge-sensitive trigger that determines when the D-input is read (or, put another way, it determines when the outputs are allowed to change). This means that the D-input is synchronous and is only read on the clock signal's positive edge for the device in Figure 2a or on the clock signal's negative edge for the device in Figure 2b.

The D-type FF's operation is actually very simple. The logic level on the D-input at the instant the device is triggered or *clocked* is copied to the Q-output. Put another way, Q *follows* D after the clock. The device's truth table is shown in Table 1.

Table 1

D	Q_{n+1}	$\overline{Q}_{n+1}$	Description
0	0	1	Q is reset (after the FF is clocked)
1	1	0	Q is set (after the FF is clocked)

There are two interesting things to note about Table 1. First, the outputs are denoted as Q_{n+1} and $\overline{Q}_{n+1}$. The "n+1" is used widely in digital electronics to indicate that this is the output's condition after it has been clocked. (On the same basis, the labels Q_n and $\overline{Q}_n$ refer to outputs before they are clocked.)

The second thing to note about Table 1 is that it doesn't give any indication about whether the flip flop's clock is positive or negative edge-triggered. For this information, you must refer to the device's schematic symbol or its data sheet.

The operation of a D-type FF can be described using a timing diagram. Figure 3 below is a timing diagram showing the outputs of a positive edge-triggered D-type FF for given data on the D-input. Notice that, whenever the clock changes from a logic-0 to a logic-1, the data on the D-input at that instant is transferred to the Q-output.

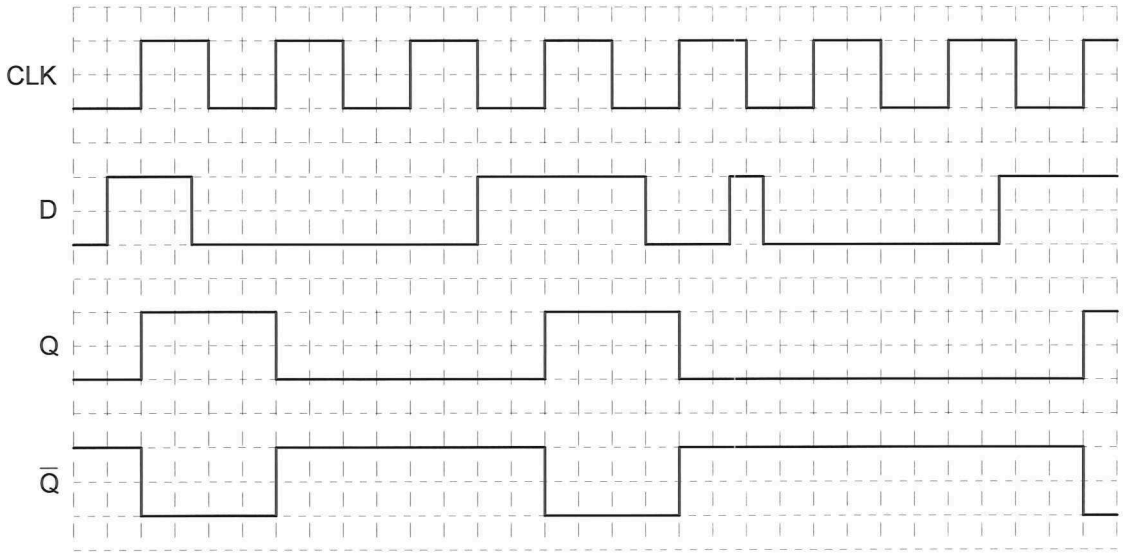


Figure 3 *A timing diagram showing the logic levels on the outputs of a positive edge-triggered D-type FF for given data on the D-input*

To help explain how to draw a timing diagram for clocked D-type flip flops, the procedure is demonstrated by showing how the timing diagram in Figure 3 on the previous page was drawn.

1. Write down the truth table of the D-type flip flop in free space close to the timing diagram.

D	Q_{n+1}	$\overline{Q}_{n+1}$
0	0	1
1	1	0

2. Identify whether the D-type FF is positive or negative edge-triggered and mark the appropriate edges of the clock on the diagram. Figure 4 below shows the positive edges marked in this example.

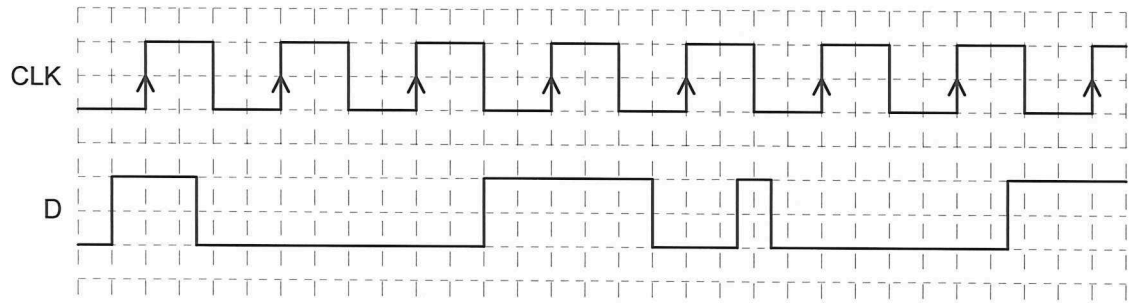


Figure 4

3. Project these edges down to *Q* through the data on *D* and identify the logic level that are on the D-input when the triggering edge occurs. See Figure 5 below.

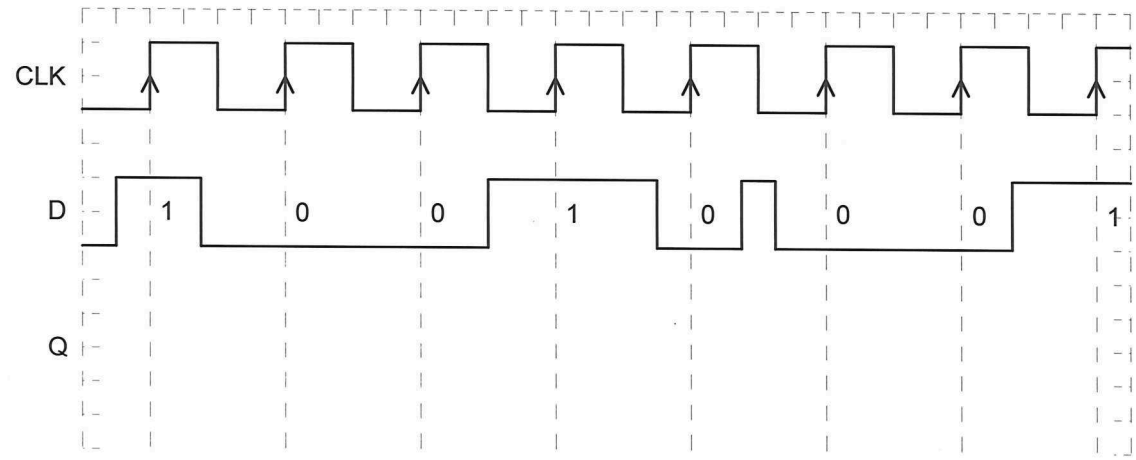


Figure 5

4. Use the D-type flip flop's truth table to determine the logic level on the Q-output then draw Q making sure that changes only occur in time with the triggering edges. See Figure 6 below. Note: If a starting logic level is not given for Q and $\overline{Q}$ then assume a logic level (it doesn't matter which one).

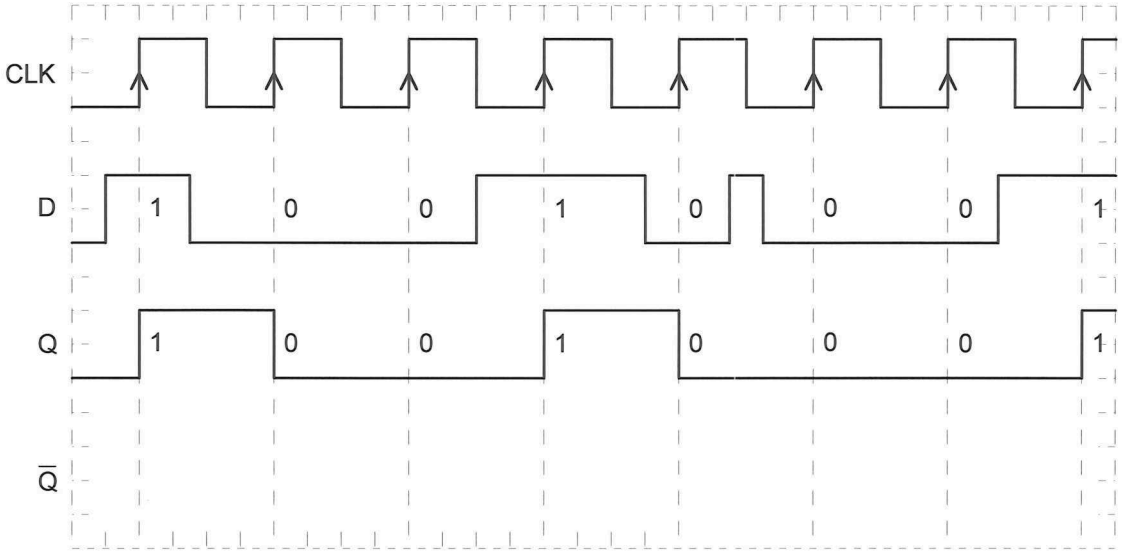


Figure 6

5. Once all the logic levels for the Q-output are drawn, draw the logic levels for the $\overline{Q}$-output by simply drawing the Q-output's signal upside down. See Figure 7 below.

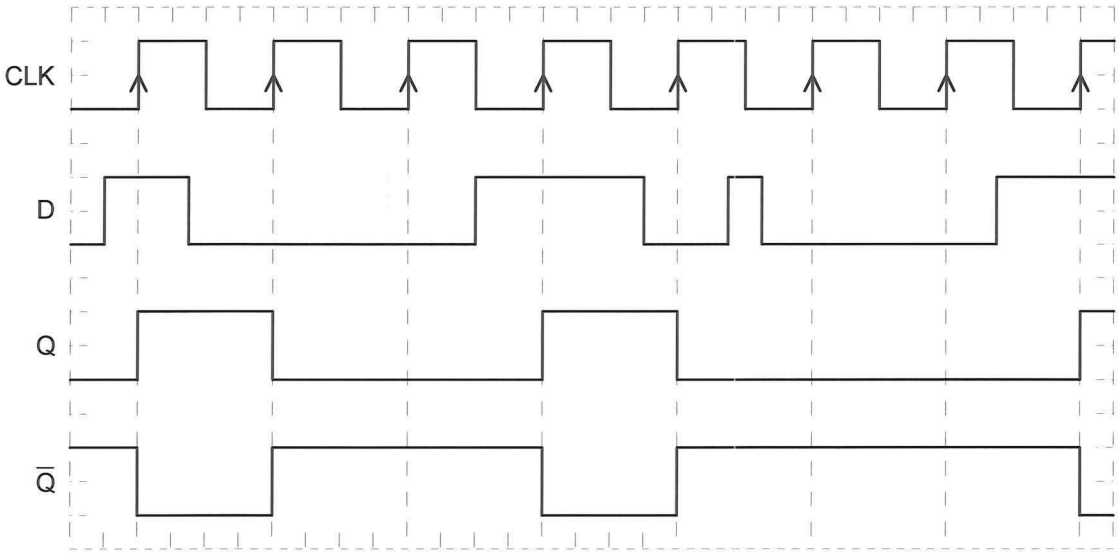
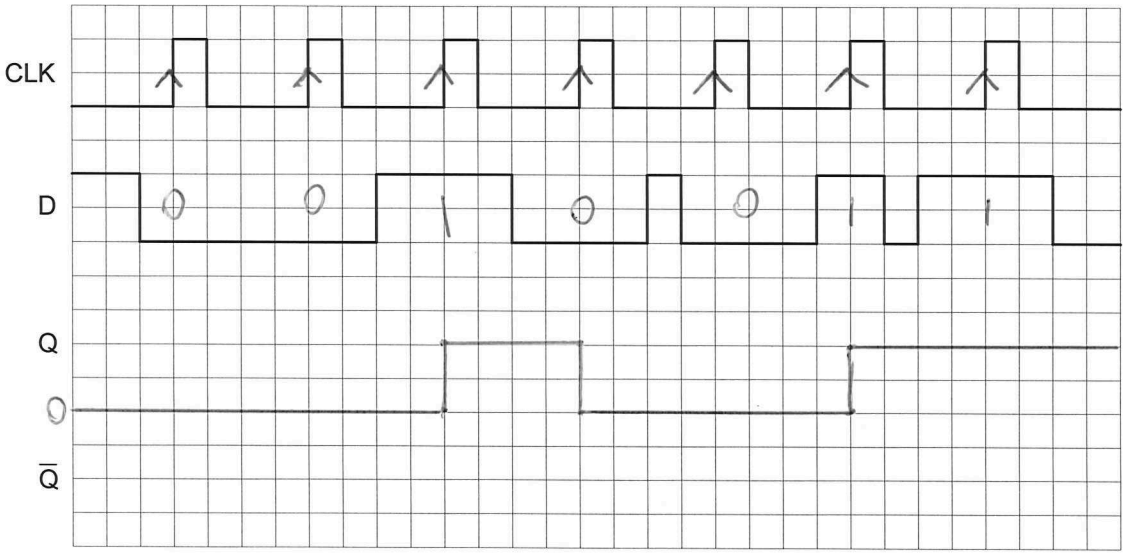


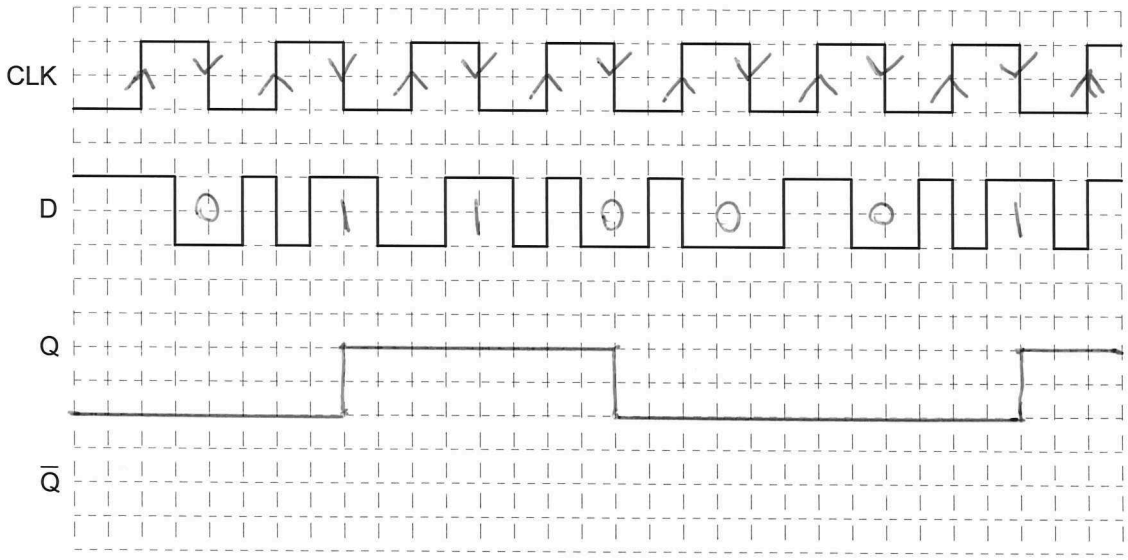
Figure 7

Practise drawing timing diagrams for D-type flip flops by trying the following questions.

1. Complete the timing diagram below for a positive edge-triggered D-type FF.



2. Complete the timing diagram below for a negative edge-triggered D-type FF.



D-type flip flops with preset and clear inputs

The D-type flip flops inside common commercially available TTL and CMOS ICs usually have a P-input and/or a C-input. An example is shown in Figure 8 below.

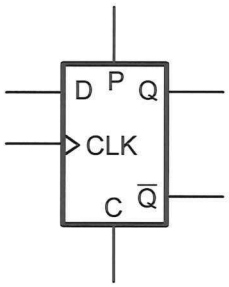


Figure 8 *A positive edge-triggered D-type FF with active high preset and clear inputs*

The letter *P* stands for *preset* and *C* stands for *clear*. The names of these inputs describe their function. When the P-input is asserted (using logic-1 for the example in Figure 8) the Q-output becomes a logic-1 and when the C-input is asserted (again, using a logic-1 in this example), the output becomes a logic-0. This is shown in Figure 9 below.

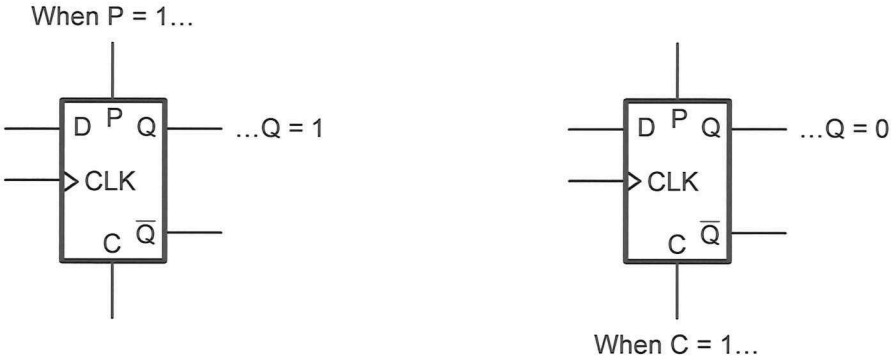


Figure 9 *When P is asserted Q becomes a logic-1 and when C is asserted Q becomes a logic-0*

Although the D-input can be used to make the Q-output either a logic-0 or a logic-1, there are two very important differences between D-input and the *P* and *C* inputs. First, the *P* and *C* inputs are asynchronous. That is, they cause the Q-output to change immediately (not withstanding the propagation delay), they don't have to wait for the next appropriate transition of the clock to cause a change on the FF's output.

Second, the *P* and *C* inputs over-ride the D-input and the clock. This means that, when either the *P* or *C* inputs are being asserted, the device does not read its D-input even if the device is clocked. For this reason, the *P* and *C* inputs are sometimes referred to as *master preset* and *master clear.*

D-type flip flops with active low preset and clear inputs are also available. The schematic symbols of the device and the basic operation of these inputs is shown in Figure 10 below.

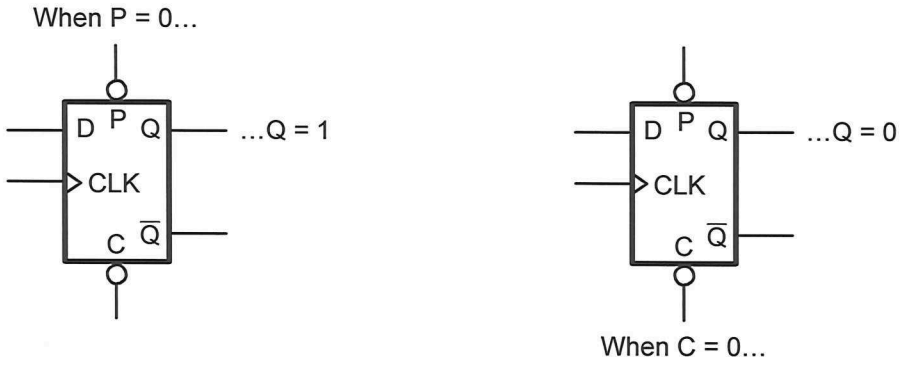


Figure 10 *A positive edge-triggered D-type FF with active low preset and clear inputs*

The operation of the preset and clear inputs on a D-type FF can also be described using a timing diagram. Figure 11 below shows a timing diagram for a positive edge-triggered D-type with active high preset and clear inputs for given data on the inputs.

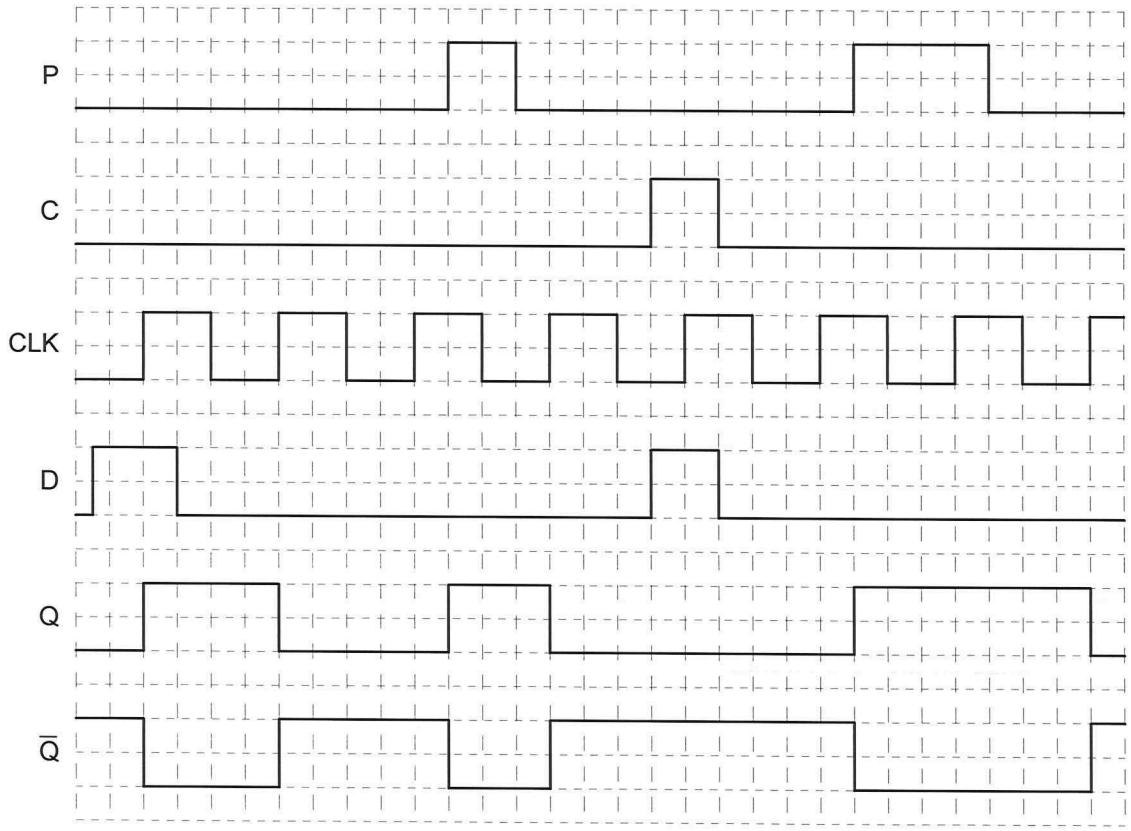


Figure 11

The procedure for drawing the timing diagram of a D-type FF with preset and clear inputs is basically the same as the procedure described earlier with a few extra steps at the beginning. These extra steps are demonstrated by showing you how the timing diagram in Figure 11 was developed.

1. Determine whether the *P* and *C* inputs are active high or active low.

2. Identify any instances in the timing diagram when the *P* and *C* inputs are asserted. See Figure 12 below.

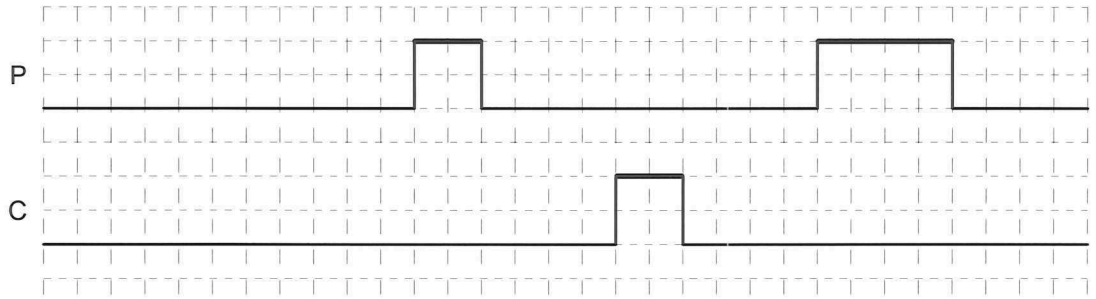


Figure 12

3. Project the leading edge of the asserted *P* and *C* inputs down to *Q*. See Figure 13 below.

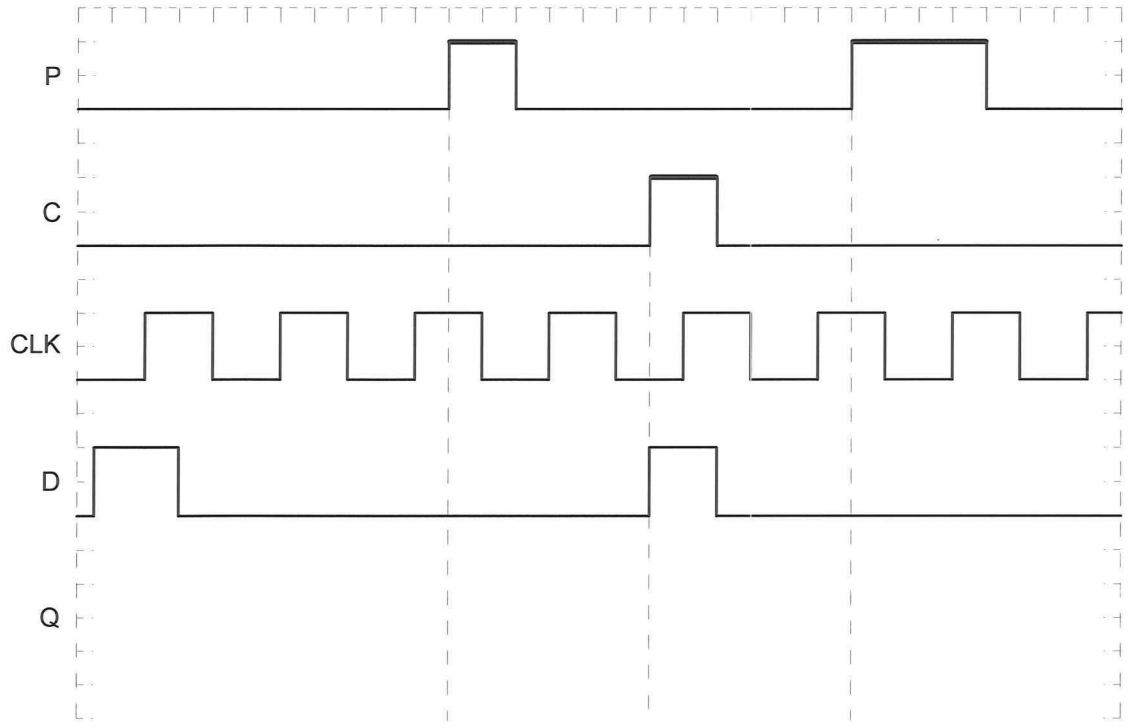


Figure 13

4. Identify which edge of the clock the device triggers off and mark these as normal. Figure 14 below shows the positive edges marked in this example.

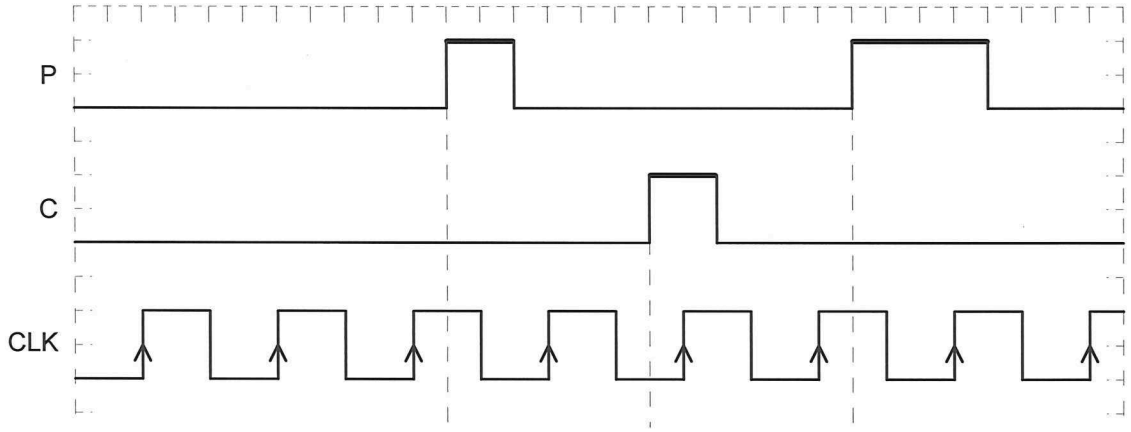


Figure 14

5. Project these edges down to *Q* **ignoring** any that occur when either *P* or *C* are asserted (remember, *P* and *C* override the D-input and clock). See Figure 15 below.

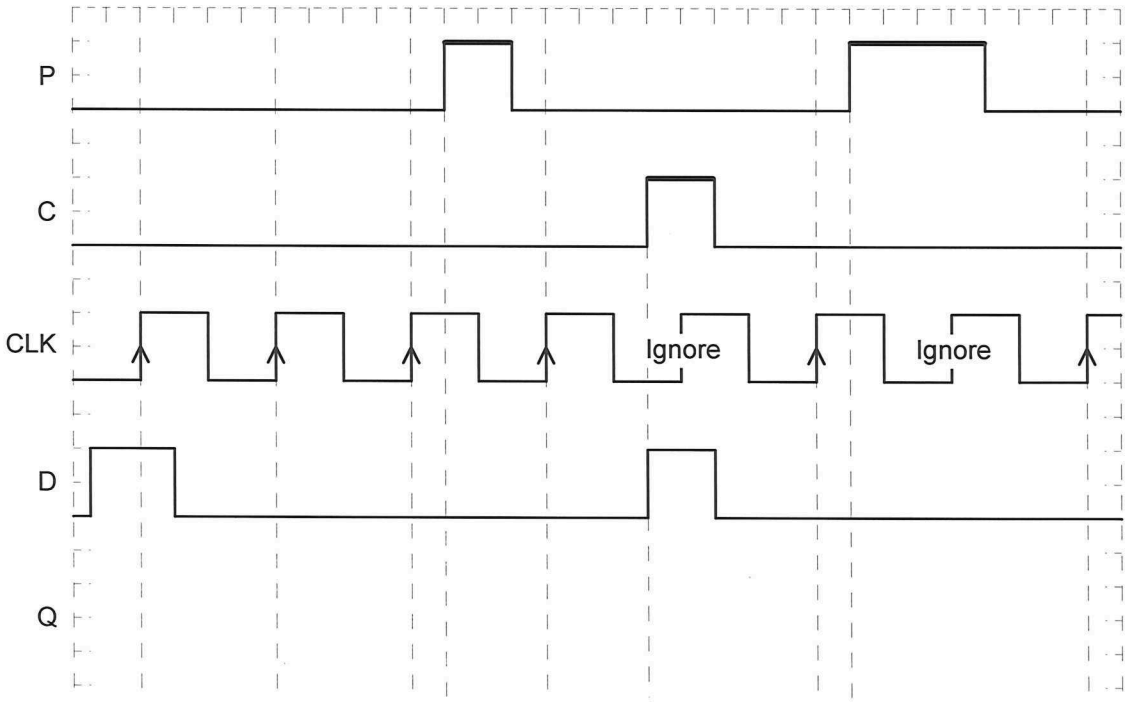


Figure 15

Copyright: Not to be reproduced without permission

6. Draw the appropriate logic level of Q for all instances when P and C are asserted starting from the first moment that they're asserted to the first clock edge that occurs after P and C are no-longer asserted. See Figure 16 below.

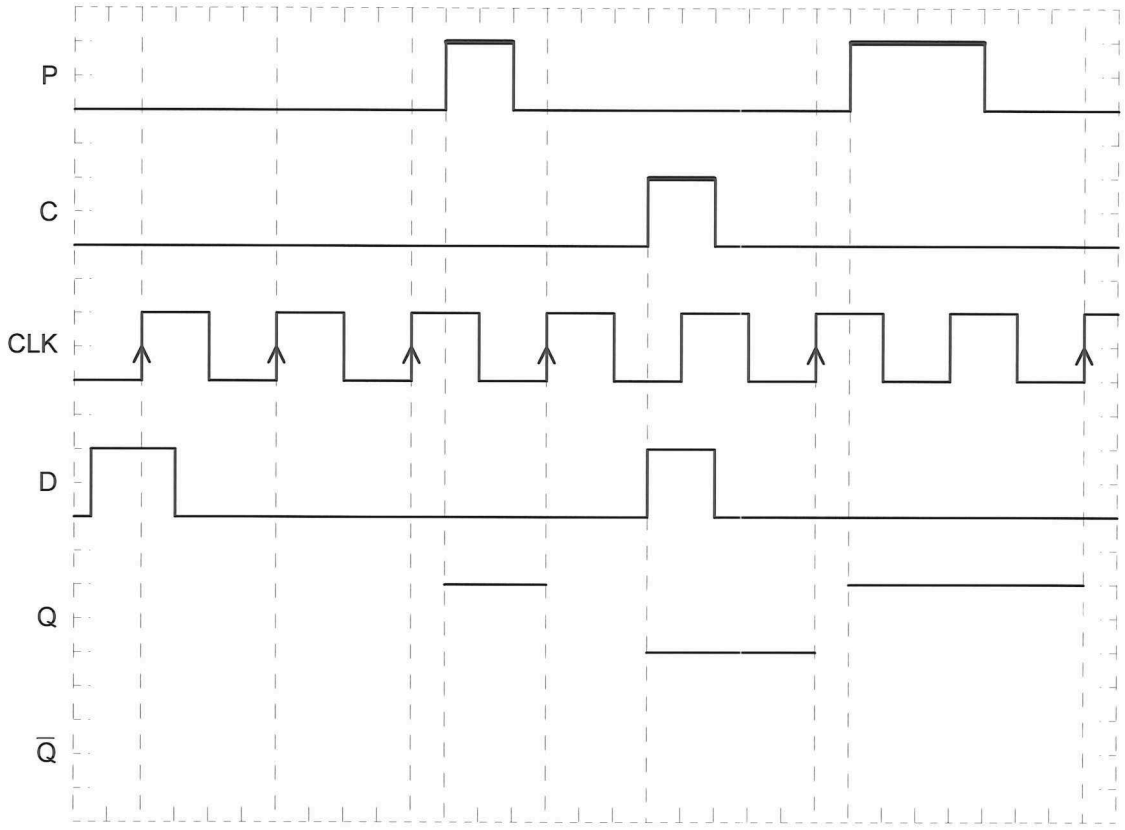


Figure 16

7. Complete the timing diagram as normal by identifying the logic level on the D-input when the device is clocked, drawing the signal for Q followed by the signal for $\overline{Q}$ (see Figures 17 to 19).

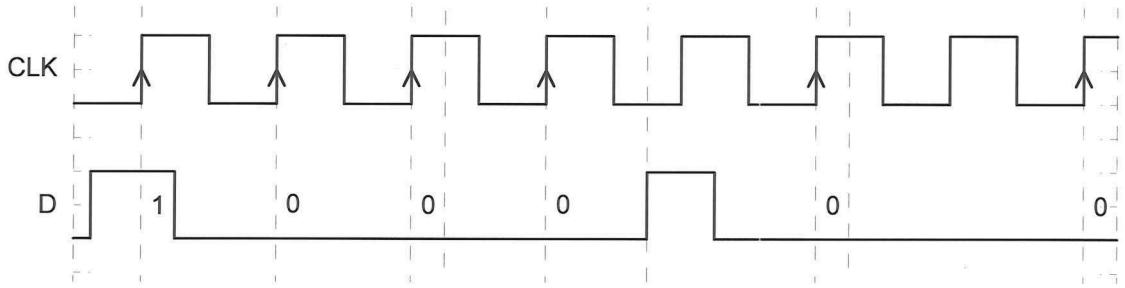


Figure 17

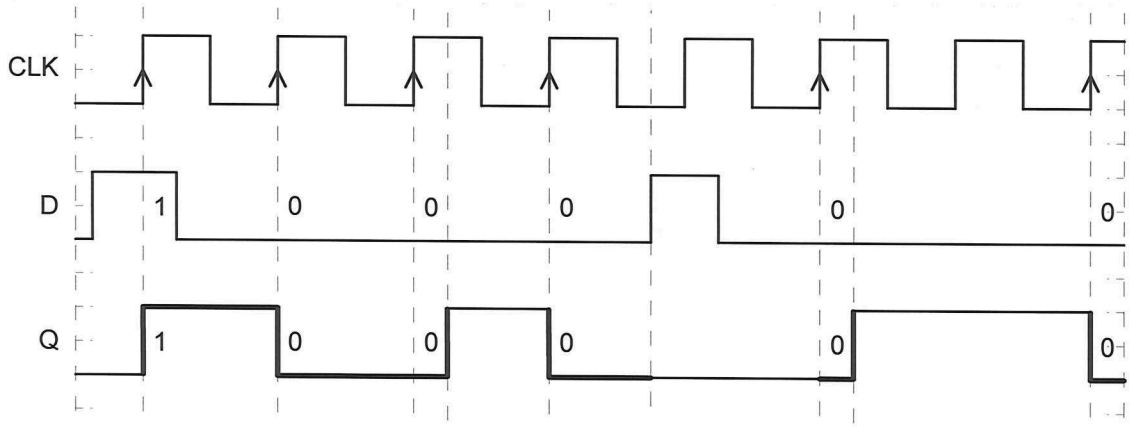


Figure 18

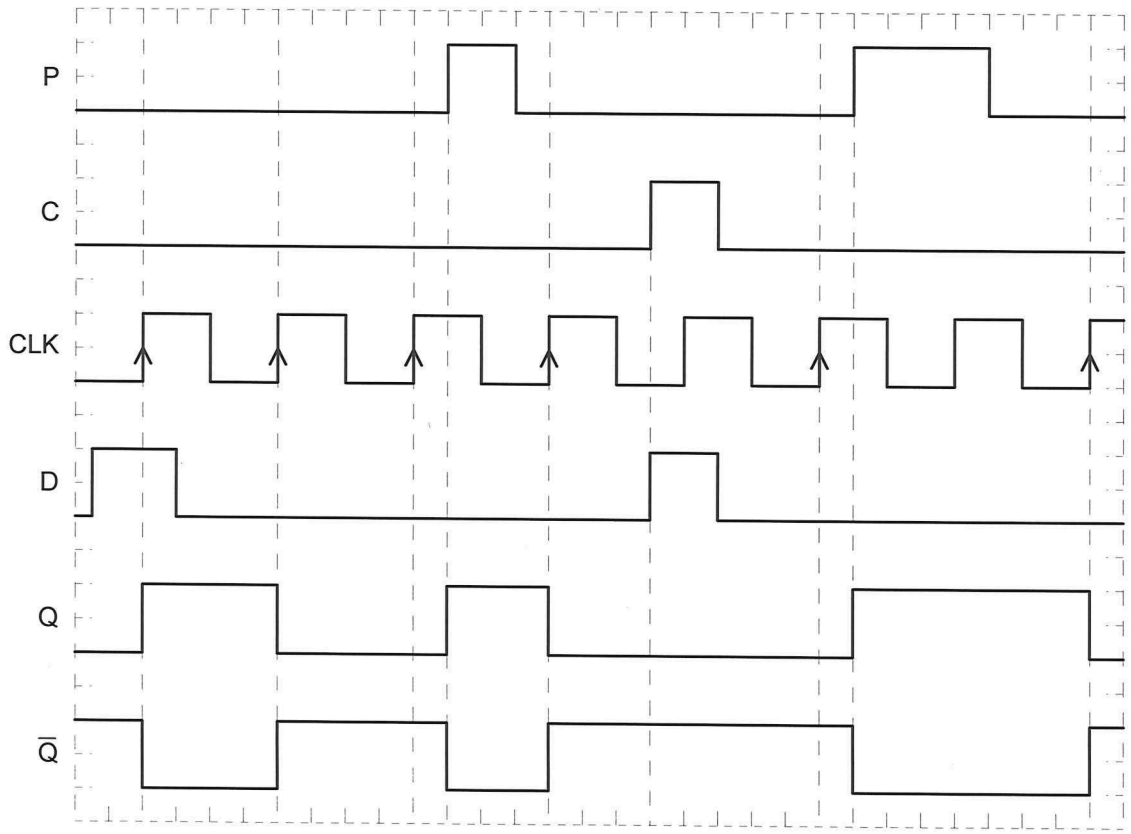


Figure 19

Practise drawing timing diagrams for D-type flip flops with preset and clear inputs by trying the following questions.

1. Complete the timing diagram below for the D-type FF shown in Figure 20.

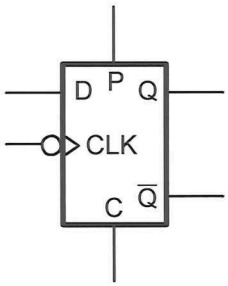
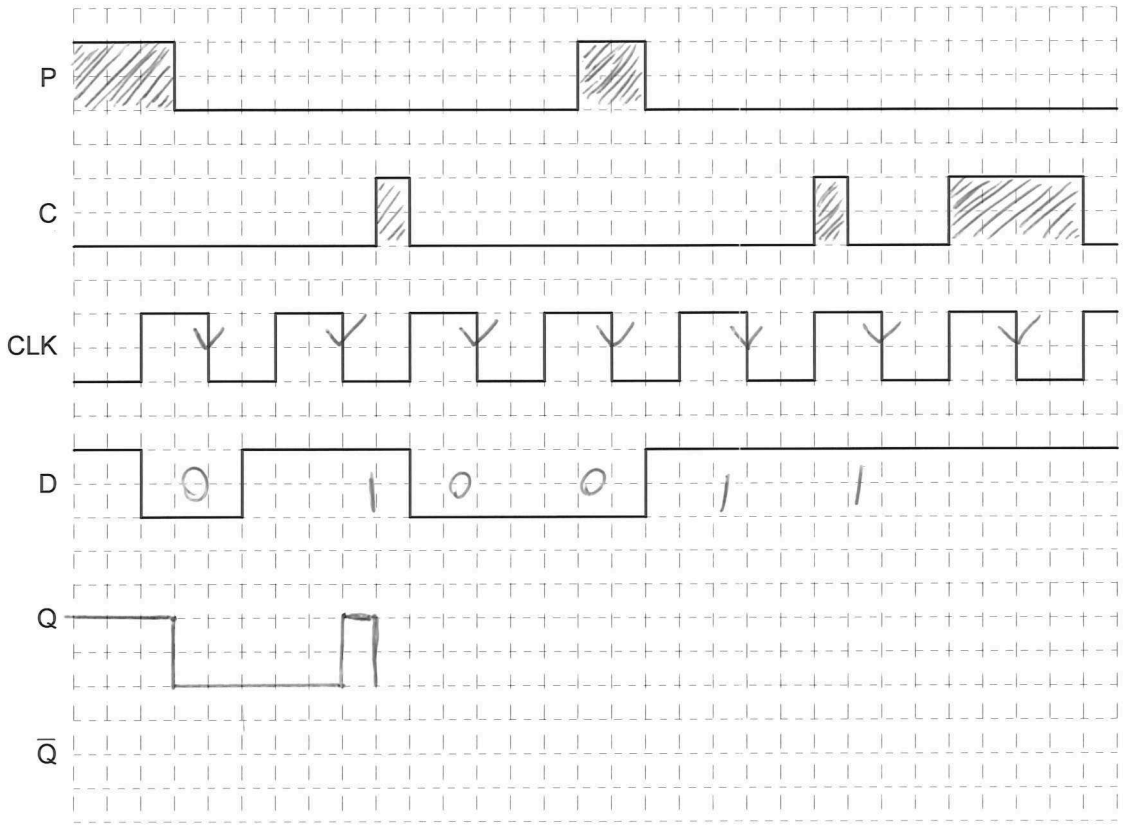


Figure 20



2. Complete the timing diagram below for the D-type FF shown in Figure 21.

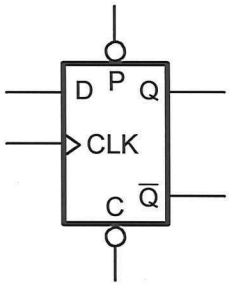
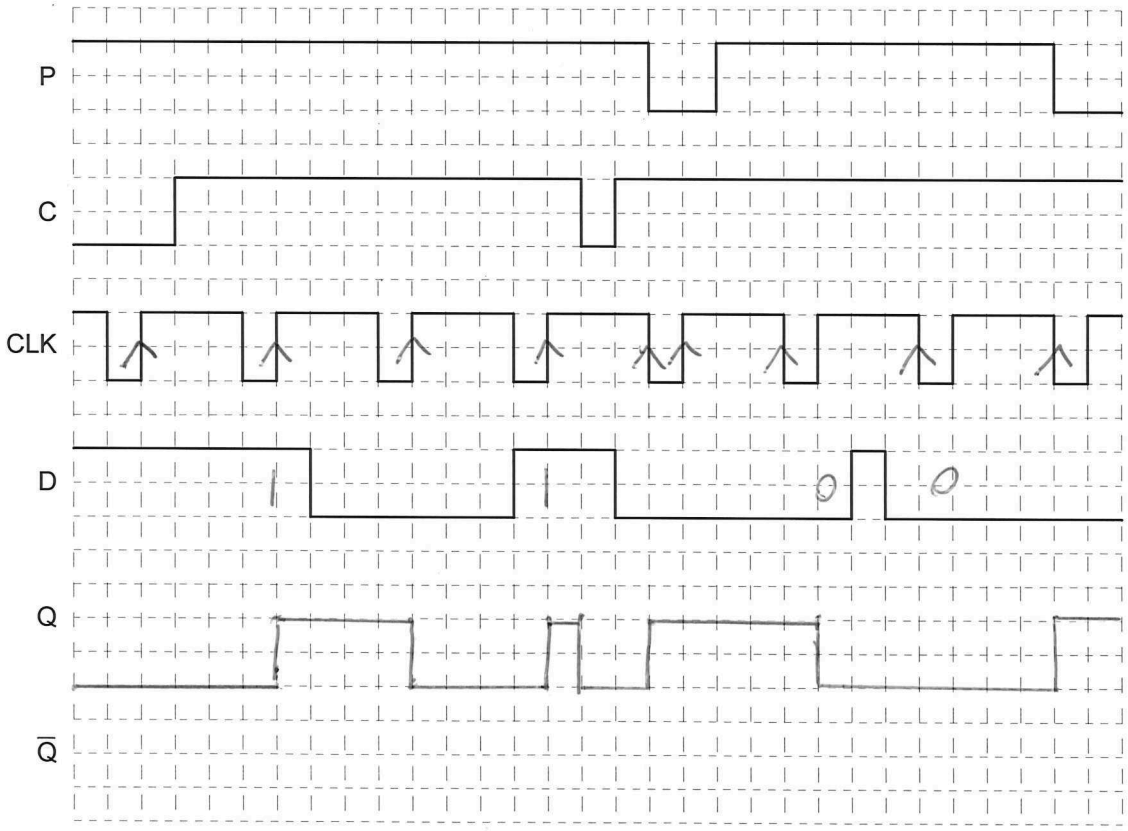


Figure 21



The need for preset and clear inputs

There is a very good reason for having preset and clear inputs on D-type FFs (and J-K FFs). There are times when the logic level on the output of sequential logic devices and circuits need to be over-ridden.

There are many reasons and applications for this as you'll learn. For example, the ability to over-ride an output is often needed when the circuit is first turned on. To explain, D-type FFs are first powered up, the Q-output does not necessarily always go the same logic level. That is, the Q-output of some flip flops might usually start as logic-0 but for others it might be logic-1 instead. Importantly, the fact that the flip flops' initial outputs are not determined by circuit can cause problems to the system. The preset and clear inputs are used to prevent these problems with extra circuitry that asserts one or the other from the moment the equipment is turned on. Doing so forces the flip flops' outputs to be an appropriate logic level from the very beginning.

Figure 22 shows how the output of a D-type FF can be forced to start as a logic-0 on power up using a simple RC circuit connected to *C*.

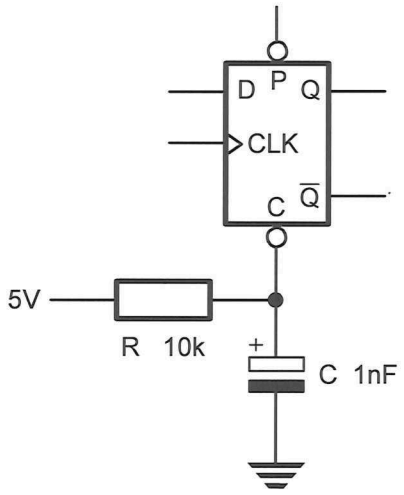


Figure 22 *A simple RC circuit is used to assert the active low C-input of a D-type FF every time the circuit is powered up*

If you examine the RC circuit closely, you'll notice that it is a series circuit connected between 5V and ground. This means that capacitor charges up whenever power to the circuit is turned on and discharges whenever power is turned off. The resistor and capacitor values are chosen so that this happens quickly but not too quickly (in this case, 50μs). That being the case, when the circuit is first turned on, the fully discharged capacitor begins to charge up to 5V. As we know, this occurs exponentially and so for a short time the potential difference across the capacitor is a voltage close to 0V. This means that, initially, a logic-0 is applied to the clear input asserting it and resetting the Q-output on power up.

Importantly, if the *P* and *C* inputs to a flip flop are not required by the circuit they shouldn't just be left not connected to anything. The voltage on unconnected inputs can *float* (that is, drift up and down over several volts) and this can randomly assert the *P* and *C* inputs, setting or resetting the device and disrupting the circuit's operation. To prevent this problem, unused *P* and *C* inputs on flip flops should be hard-wired to the logic level that doesn't assert them, preventing them from floating.

Applications for D-type flip flops

D-type flip flops with preset and clear inputs can be used instead of RS flip flops for all RS FF applications including the latch. This is because the *P* and *C* inputs of the D-type FF act in exactly the same way as the *S* and *R* inputs of the RS FF. To be more specific, if a NOR RS FF is needed then a D-type FF with active high *P* and *C* inputs is used and if a NAND RS FF is needed then a D-type FF with active low *P* and *C* inputs is used. In both cases, the clock input should be hard-wired to an appropriate logic level to prevent the FF from being accidently clocked when neither *P* nor *C* are asserted.

Another application for the D-type FF is a digital *frequency divider*. The circuit for this is shown in Figure 23 below.

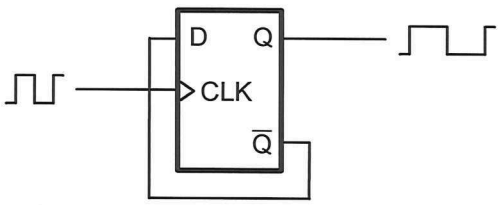


Figure 23 *A D-type FF wired to operate as a digital signal frequency divider (÷ by 2)*

As you can see, the only input to the circuit is a clock signal connected to the clock input. There is no need for an external source of data because the D-input is hard-wired to the $\overline{Q}$-output. In other words, the $\overline{Q}$-output is providing the data for the D-input. This means that, every time the device is clocked, Q becomes the same logic level on $\overline{Q}$ before the clock (that is, $Q_{n+1} = \overline{Q}_n$) and $\overline{Q}$ becomes the same as the logic level on Q before the clock (that is, $\overline{Q}_{n+1} = Q_n$).

For example, if the outputs start out as $Q = 0$ and $\overline{Q} = 1$, then after the clock they change to $Q = 1$ and $\overline{Q} = 1$. In other words, the outputs alternate between logic-0 and logic 1 and they are said to *toggle*.

Although the previous discussion explains the toggling action of the circuit's outputs, a timing diagram is needed to explain why it is called a frequency divider. Figure 24 below shows the timing diagram for the circuit in Figure 23. It is assumed that Q starts as a logic-0. Also $\overline{Q}$ is not shown separately because D is the same as $\overline{Q}$ (they're connected together).

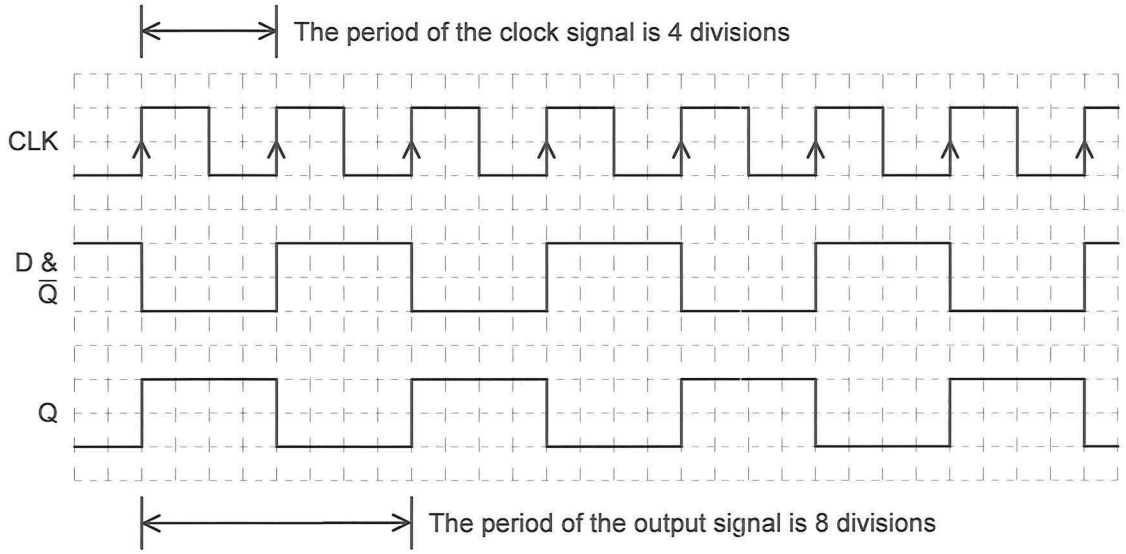


Figure 24

The timing diagram shows that the clock signal's period in this case is four divisions and, as the Q-output can only change on the positive edge of the clock, its period is eight divisions. This is a doubling of the period which means a halving of the frequency (remember, $f = \frac{1}{P}$).

D-type flip flops (and J-K flip flops) are also used to make other important sequential logic circuits including *counters* and *shift registers*. As these devices are dealt with later in the workbook they're not discussed any further here.

Skill Practice 1

Practise testing and measuring sequential logic circuits (D-type flip flops)

This exercise is practise for the sorts of skills you may be required to perform in a practical test. Remember, in any practical tests you will be working alone so make sure that you can perform all the steps. It should take you about 35 minutes to complete this exercise.

Equipment

- Digital trainer
- logic probe
- 74LS74
- two BNC to alligator-clip leads
- hook-up wire (terminated)
- IC extractor

Remember:

Follow TAFE NSW WHS guidance at all times!

Work tasks

1. Read your WHS responsibilities at the top of the form below. Then conduct a WHS risk assessment and record your findings in the space provided.

Responsibilities of students under the Model WHS Act: s28

- Take reasonable care for your own health and safety by working safely at all times
- Take reasonable care to ensure that your acts or omissions don't put the health and safety of others at risk
- Follow all TAFE NSW WHS guidance and comply with all reasonable instructions from TAFE NSW staff to assist them in complying with the TAFE NSW WHS requirements
- In addition to the above, you must:
  - use and maintain machinery, tools and all other equipment properly and safely
  - ensure that your work area is free of hazards
  - notify a TAFE NSW staff member of actual or potential hazards
  - wear/use prescribed safety equipment
  - take notice of any safety signs and adhere to their instructions

Risks involved in this activity include:	**Control measures:**
■ Trip hazards (eg students bags) ■ Objects dropped on feet (while equipment is taken to and from benches).	■ Move bags and other objects from walkways ■ Plan lifting of equipment
Others: _____ _____ _____	Other: _____ _____ _____

My signature here indicates that I have read and understand my responsibilities under the Model WHS Act s28 (detailed above). I have also conducted a risk assessment before undertaking this activity and have identified measures to control these risks and have implemented them.

Signature: _____ Date: _____

2. Gather the equipment needed for this exercise.

3. Read the following points and make sure that you perform them when wiring all circuits:

   - Write the IC pin numbers of the devices you work with on the circuit that you're instructed to wire.
   - Handle all ICs carefully - remember ESD can damage them.
   - Ensure the power supply is connected to all ICs in the circuit.
   - When plugging hook-up wire into the breadboard to make connections to the IC's pins, use holes that are furthest away from the pin.

4. Wire the circuit of Figure 1 on the digital trainer. Pinout data for the 74LS74 is at the back of the book.

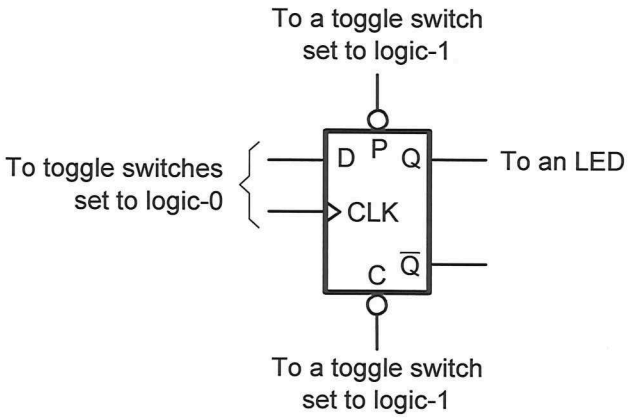


Figure 1

5. Turn on the trainer.

6. **Momentarily** throw the P-input's toggle switch to logic-0 and observe what happens to the output.

7. **Momentarily** throw the C-input's toggle switch to logic-0 and observe what happens to the output.

8. Repeat steps 6 and 7 a few times to verify your understanding of the operation of the *P* and *C* inputs.

Question 1
Are the *P* and *C* inputs active high or active low? Explain your answer in terms of what happened in steps 6 and 7.

Question 2
Are the *P* and *C* inputs asynchronous or synchronous? Explain your answer in terms of what happened in steps 6 and 7.

The teacher needs to check your work at this point...

9. **Momentarily** throw the C-input's toggle switch to logic-0 to reset the Q-output.

10. Check that the D-input's toggle switch is still logic-0.

11. **Momentarily** throw the clocks input's toggle switch to logic-1 and note the logic level on the output. Record your observation in Table 1 on the next page.

12. **Momentarily** throw the P-input's toggle switch to logic-0 to set the Q-output.

13. Repeat step 11.

14. Set the D-input's toggle switch a logic-1 and repeat step 11.

15. Leave the D-input's toggle switch set to a logic-1 and repeat step 11.

Table 1

Before step 11		After
D	**Q_n**	**Q_{n+1}**
0	0	
0	1	
1	0	
1	1	

Question 3
What happens to the Q-output when the D-input is logic-0 and the device is clocked? [**Tip:** If you're not sure, cover the Q_n column.]

Question 4
What happens to the Q-output when the D-input is logic-1 and the device is clocked?

Question 5
Is the clock input positive or negative edge-triggered? Explain your answer in terms of what happened every time you performed step 11.

The teacher needs to check your work at this point...

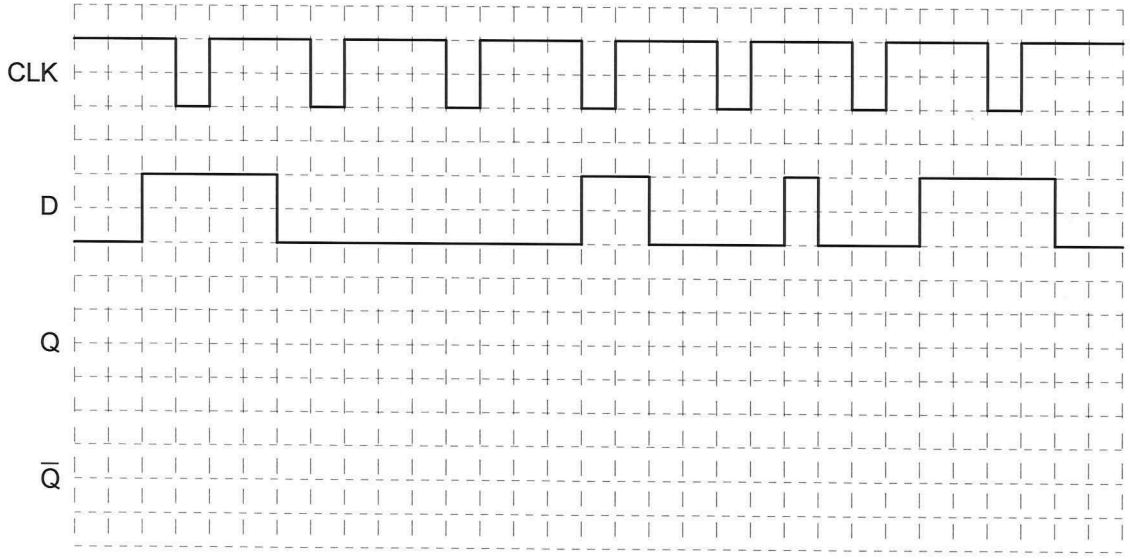
Copyright: Not to be reproduced without permission

Student notes

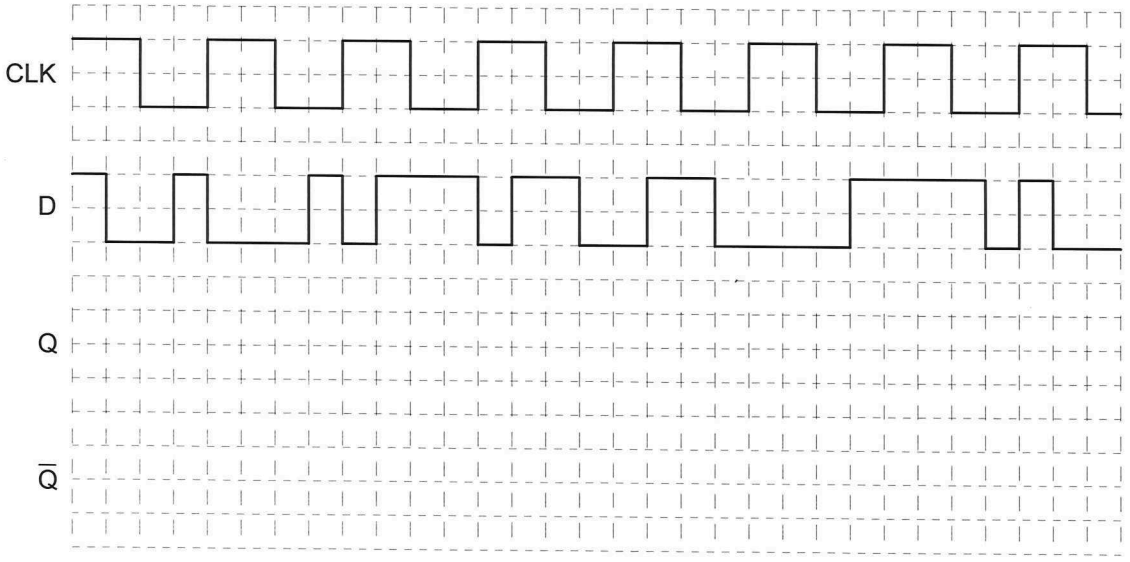
Review Questions

Answer these questions to check your understanding of what you have learnt from this chapter. Doing this will also help to prepare you for the assessments.

1. Complete the timing diagram below for a positive edge-triggered D-type FF without preset and clear inputs. Assume the Q-output starts as logic-0 (as shown).



2. Complete the timing diagram below for a negative edge-triggered D-type FF without preset and clear inputs. Assume the Q-output starts as logic-0.



Questions 3 to 8 refer to the device in Figure 1 below

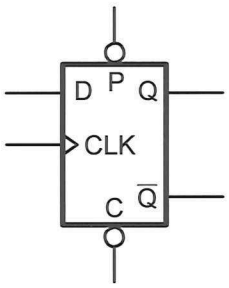


Figure 1

3. Which input or inputs are asynchronous?

- ☐ *D* only
- ☐ *CLK* only
- ☐ *P* and *C* only
- ☐ All of them
- ☐ None of them

4. Which input or inputs are synchronous?

- ☐ *D* only
- ☐ *CLK* only
- ☐ *P* and *C* only
- ☐ All of them
- ☐ None of them

5. Which input or inputs are negative edge-triggered?

- ☐ *D* only
- ☐ *CLK* only
- ☐ *P* and *C* only
- ☐ All of them
- ☐ None of them

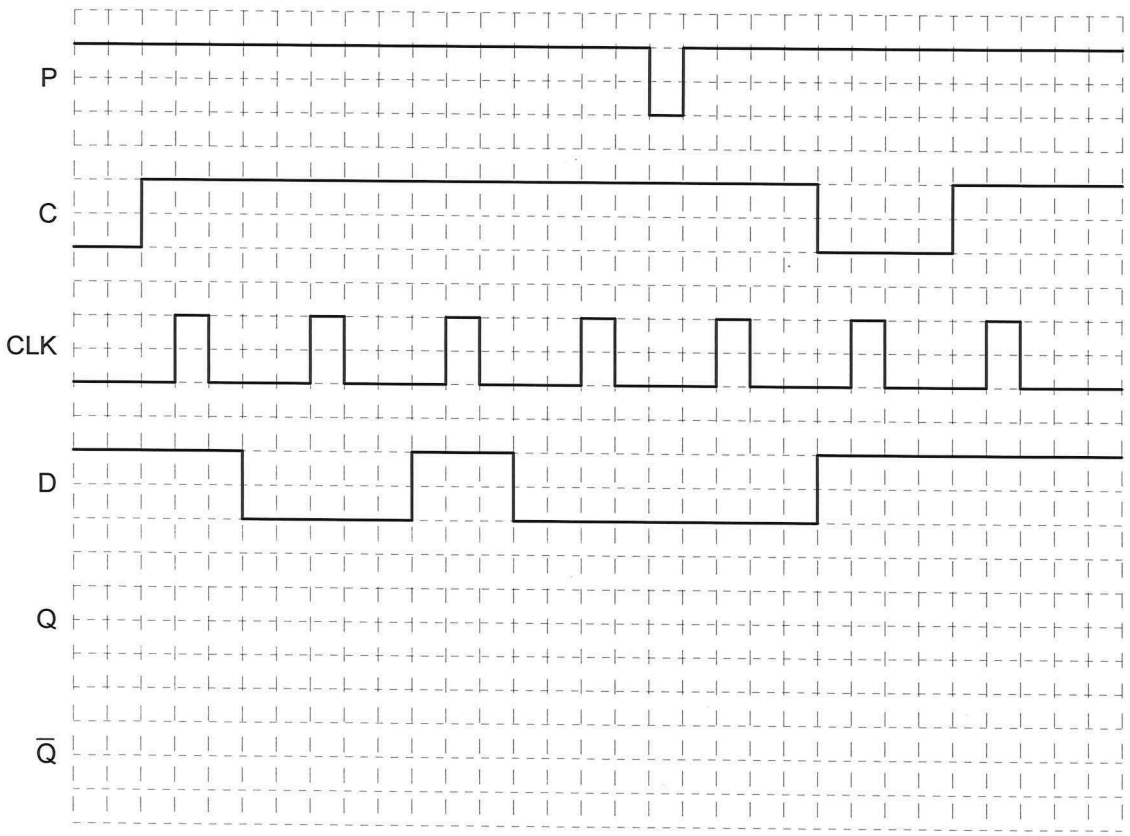
6. Which inputs are active low?

- ☐ *D* only
- ☐ *CLK* only
- ☐ *P* and *C* only
- ☐ All of them
- ☐ None of them

7. Which input or inputs are edge-sensitive?

- ☐ *D* only
- ☐ *CLK* only
- ☐ *P* and *C* only
- ☐ All of them
- ☐ None of them

8. Complete the timing diagram below for the device in Figure 1.



 Copyright: Not to be reproduced without permission

9. Complete the timing diagram below for the device in Figure 2 below.

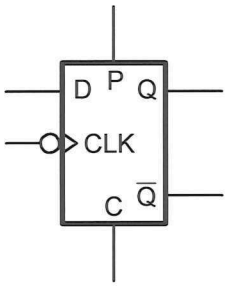
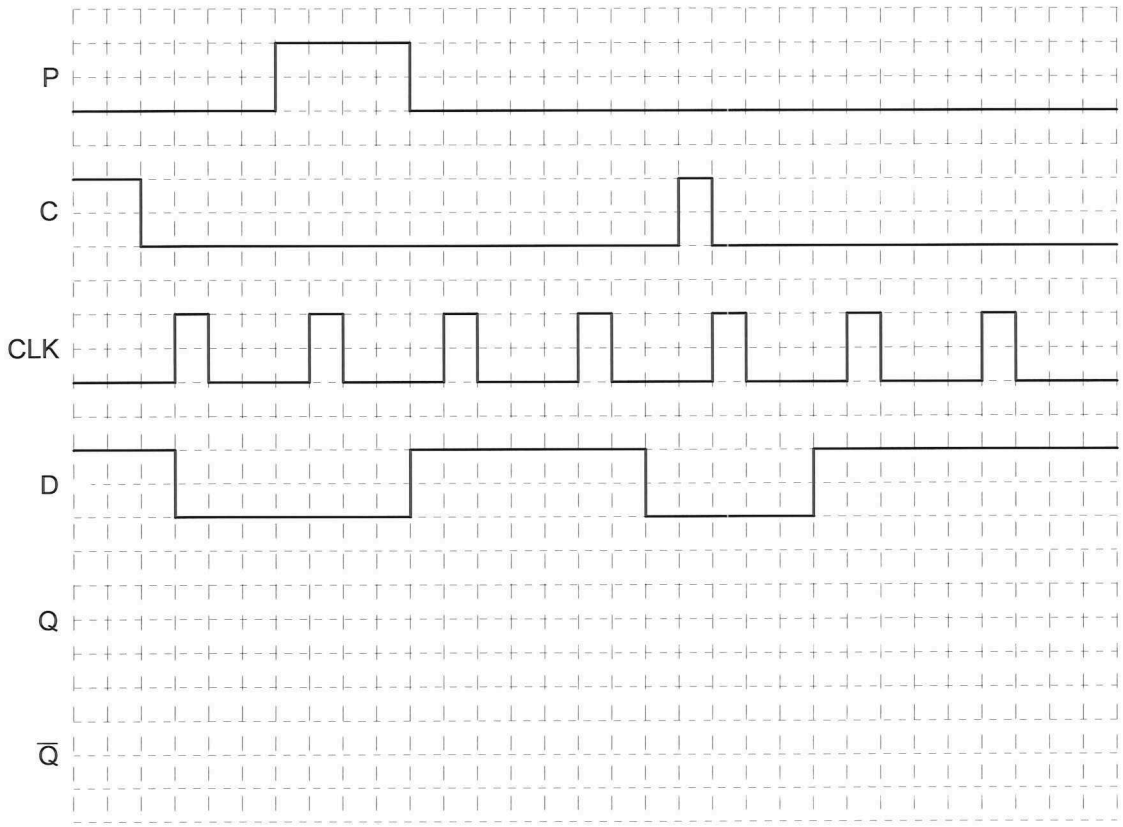


Figure 2



10. List four applications for D-type flip flops.

Questions 11 and 12 refer to the circuit in Figure 3 below

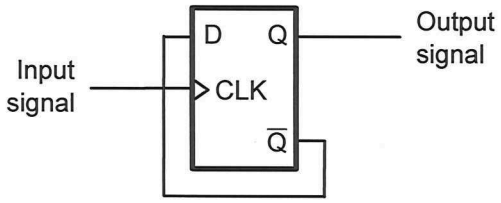


Figure 3

11. What is the name of the circuit in Figure 3?

12. What is the relationship between the input and output signals?

Copyright: Not to be reproduced without permission

Student notes

Section 2 The J-K flip flop

Purpose To develop your understanding of the J-K flip flop and to further develop your ability to interpret the manufacturers' data sheets.

Objectives At the end of this section you should be able to:

- Recognise the schematic symbol for a J-K flip flop
- Predict a JK-type flip flop's output for a given set of inputs
- Draw and/or interpret a timing diagram for a J-K flip flop (with and without preset and clear inputs) given a set of logic levels for their inputs that change over time
- Explain the problem that can be caused when preset and clear inputs on logic devices (like JK-type flip flops) are left floating and how it can be prevented
- Give applications for the J-K flip flop
- Interpret manufacturers' data for common commercially available J-K flip flops
- Use a logic probe to test and measure sequential logic circuits

Introduction

The last chapter introduced you to the D-type flip flop. This chapter finishes the topic of flip flops by introducing you to the J-K flip flop.

The J-K flip flop

If you think about the operation of the NOR RS FF you'll recall that it has a problem. If the *S* and *R* inputs are asserted at the same time the Q and $\overline{Q}$ outputs become the same logic level. This is highlighted in the truth table of the NOR RS FF in Table 1 below.

Table 1

S	R	Q	$\overline{Q}$	Description
0	0	Q	$\overline{Q}$	No change
0	1	0	1	Q is reset
1	0	1	0	Q is set
1	1	0	0	Disallowed

By definition, Q and $\overline{Q}$ must be the opposite of each other and so asserting both inputs at the same time is disallowed. However, simply disallowing this input combination ($S = 1$ and $R = 1$) is not making the most of the device. So extra circuitry is added to the NOR RS FF to create a new device that does something useful with this last input combination. Instead of being disallowed, asserting both inputs causes the outputs to toggle each time the device is clocked.

To distinguish this new device from the NOR RS FF, the inputs are labelled *J* and *K* instead of *S* and *R* and the device is called the J-K flip flop. J-K flip flops in commercially available TTL and CMOS ICs are either gated or clocked. For example, the 7473 and 7476 consist of two J-K flip flops.

Figures 1a and 1b below show the schematic symbols of clocked J-K flips flops that are positive edge-triggered and negative edge-triggered.

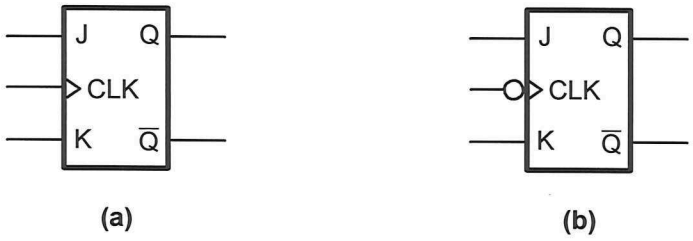


Figure 1 *Clocked J-K flip flops that are: a) positive edge-triggered; and b) negative edge-triggered*

Table 2

J	K	Q_{n+1}	$\overline{Q}_{n+1}$	Description
0	0	Q_n	$\overline{Q}_n$	No change
0	1	0	1	Q is reset
1	0	1	0	Q is set
1	1	$\overline{Q}_n$	Q_n	Outputs toggle

As with D-type FFs, the truth table gives no indication of whether the clock is positive or negative edge-triggered. For this information, you must refer to the device's schematic symbol or data sheet.

The operation of a clocked J-K FF can be described using a timing diagram. Figure 2 below is a timing diagram showing the outputs of a positive edge-triggered J-K FF for given data on the *J* and *K* inputs. It is assumed that *Q* starts as logic-0.

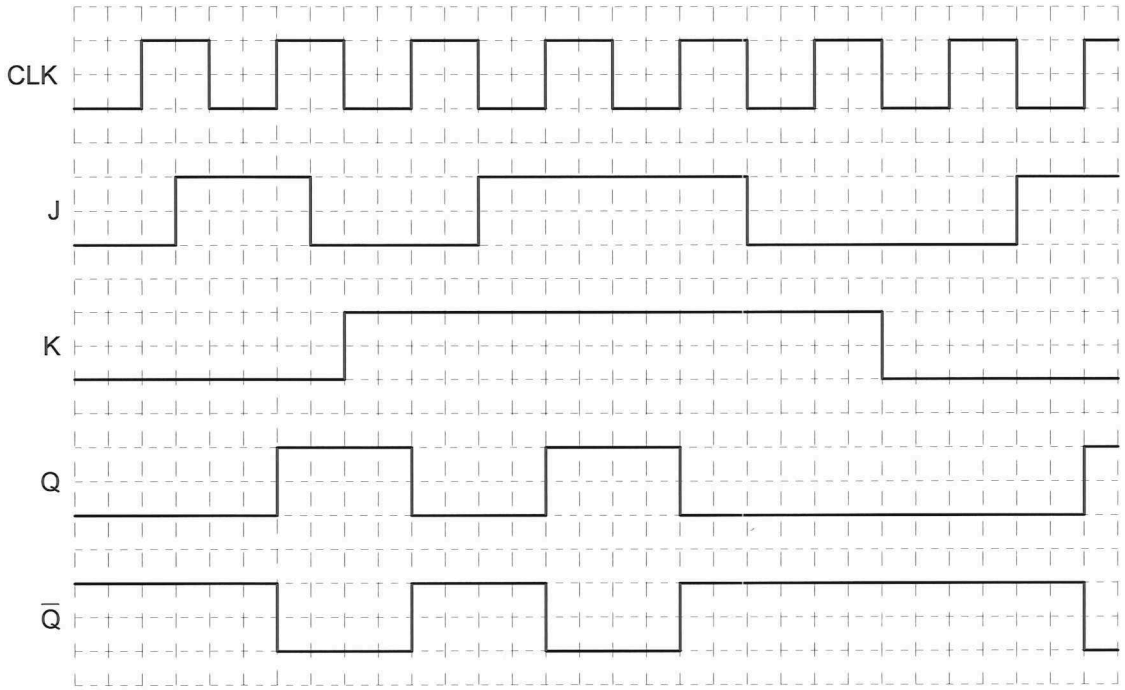


Figure 2

The procedure for developing the J-K FF timing diagrams is exactly the same for a D-type flip flops so it is not explained again here. However, the teacher will demonstrate how Figure 2 is developed and extra information is provided in Figure 3 on the next page in case you miss the lesson.

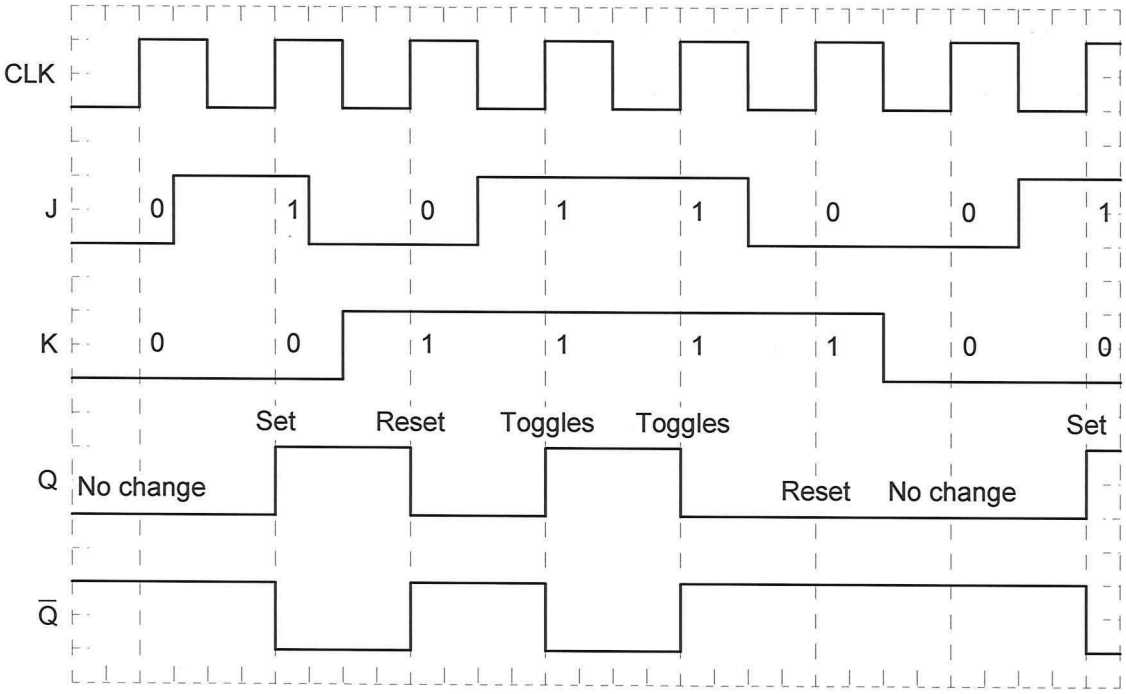
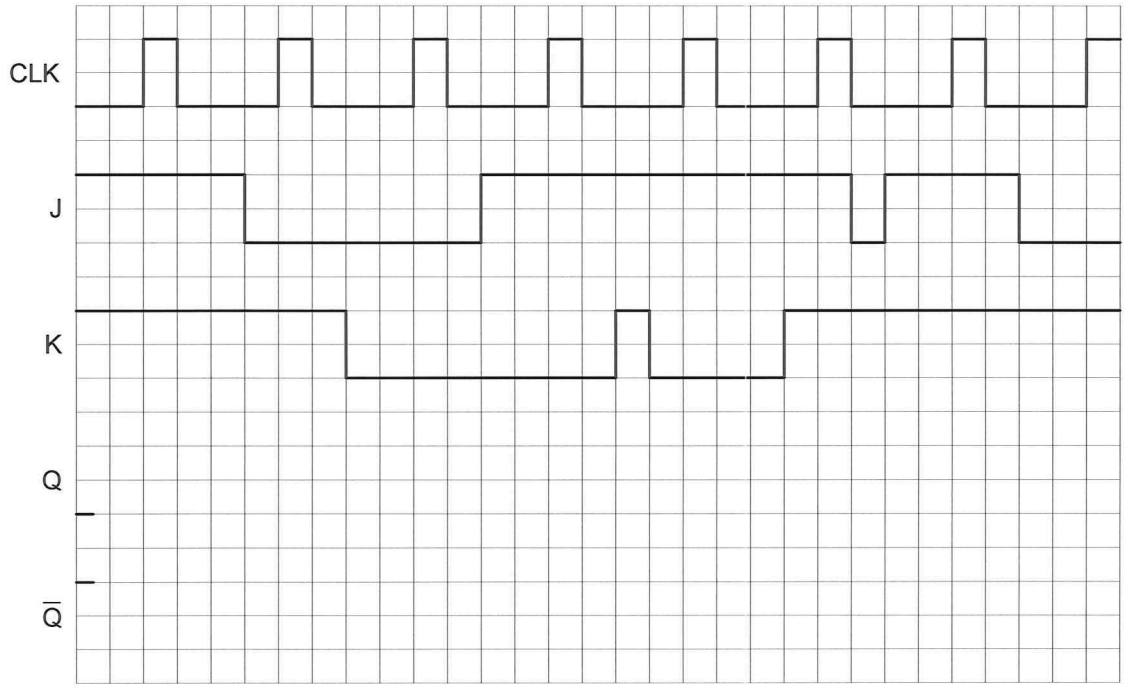

CLK
J
0
1
0
1
1
0
0
1
K
0
0
1
1
1
1
0
0
Set
Reset
Toggles
Toggles
Set
Q
No change
Reset
No change
$\overline{Q}$

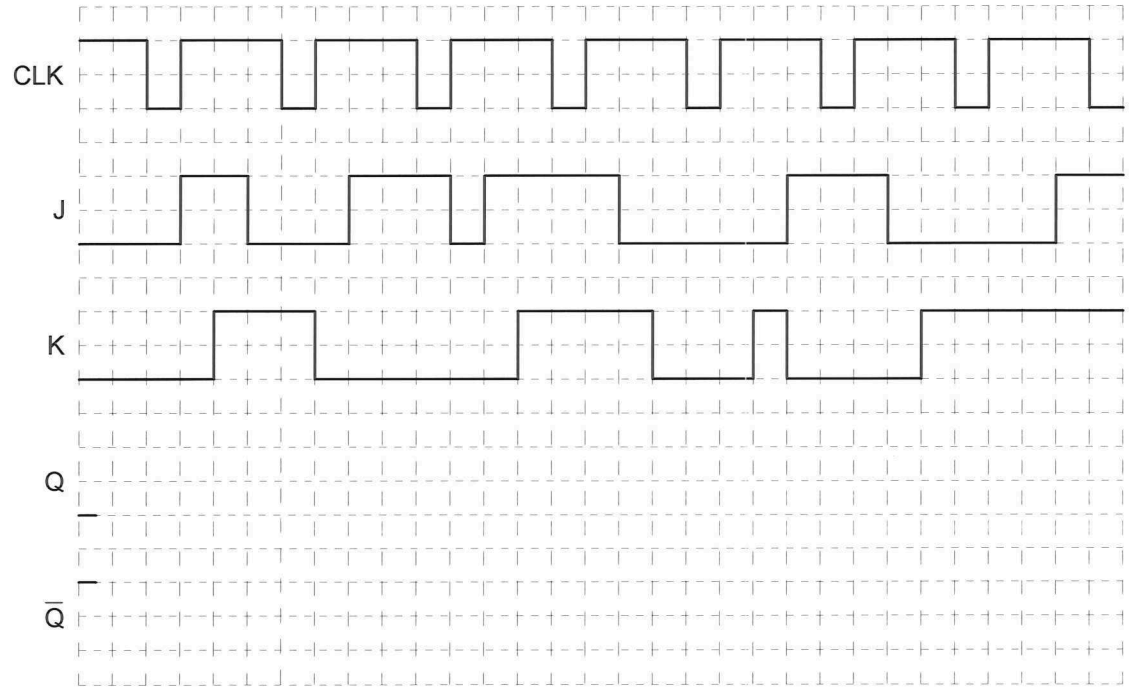
Figure 3

Practise drawing timing diagrams for J-K flip flops by trying the following questions.

1. Complete the diagram below for a positive edge-triggered J-K FF.



2. Complete the timing diagram below for a negative edge-triggered J-K FF.



J-K flip flops with preset and clear inputs

J-K flip flops inside common commercially available TTL and CMOS ICs also usually have a preset input and/or clear input. An example is shown in Figure 4 below using a positive edge-triggered J-K FF.

Figure 4 *A positive edge-triggered J-K FF with: a) active high preset and clear inputs; and b) active low preset and clear inputs*

Preset and clear inputs are provided with J-K FFs for the same reason as for D-type flip flops. Importantly, if they're not used, they shouldn't be left floating. Instead, the inputs should be hard-wired to the logic level that doesn't assert them to prevent false presetting or clearing.

Figure 5 below shows the effect of asserting active high *P* and *C* inputs of a J-K FF and Figure 6 on the next page shows the effect of asserting active low *P* and *C* inputs.

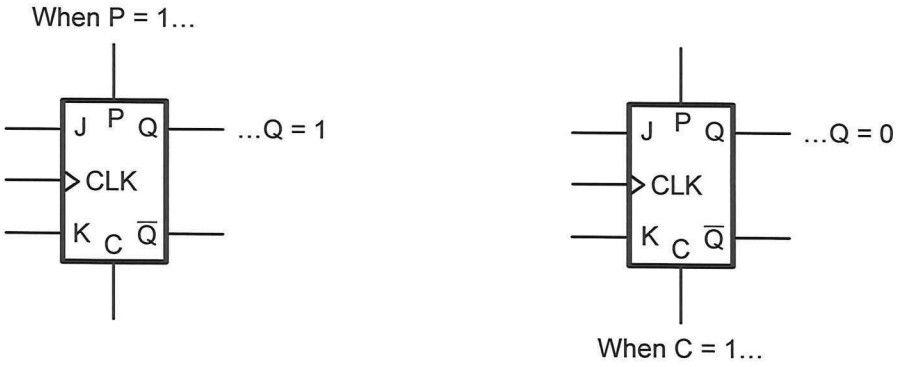


Figure 5 *The effect of asserting active high P & C of a J-K FF*

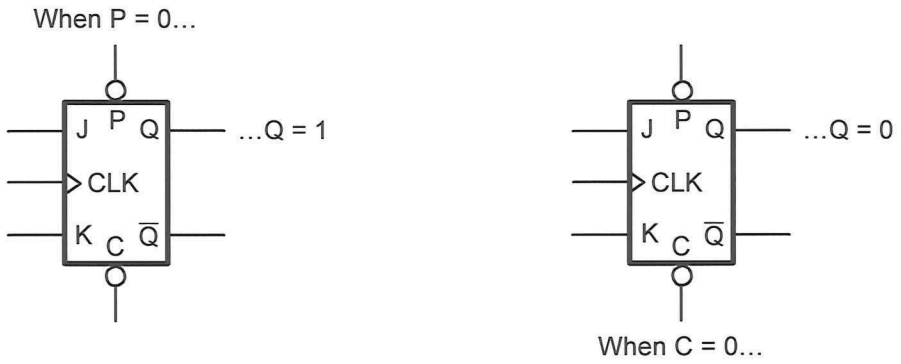


Figure 6 *The effect of asserting active low P & C of a J-K FF*

The operation of the preset and clear inputs on a J-K FF can also be described using a timing diagram. Figure 7 below is a timing diagram showing the outputs of a positive edge-triggered J-K FF with active high preset and clear inputs for given data on the inputs.

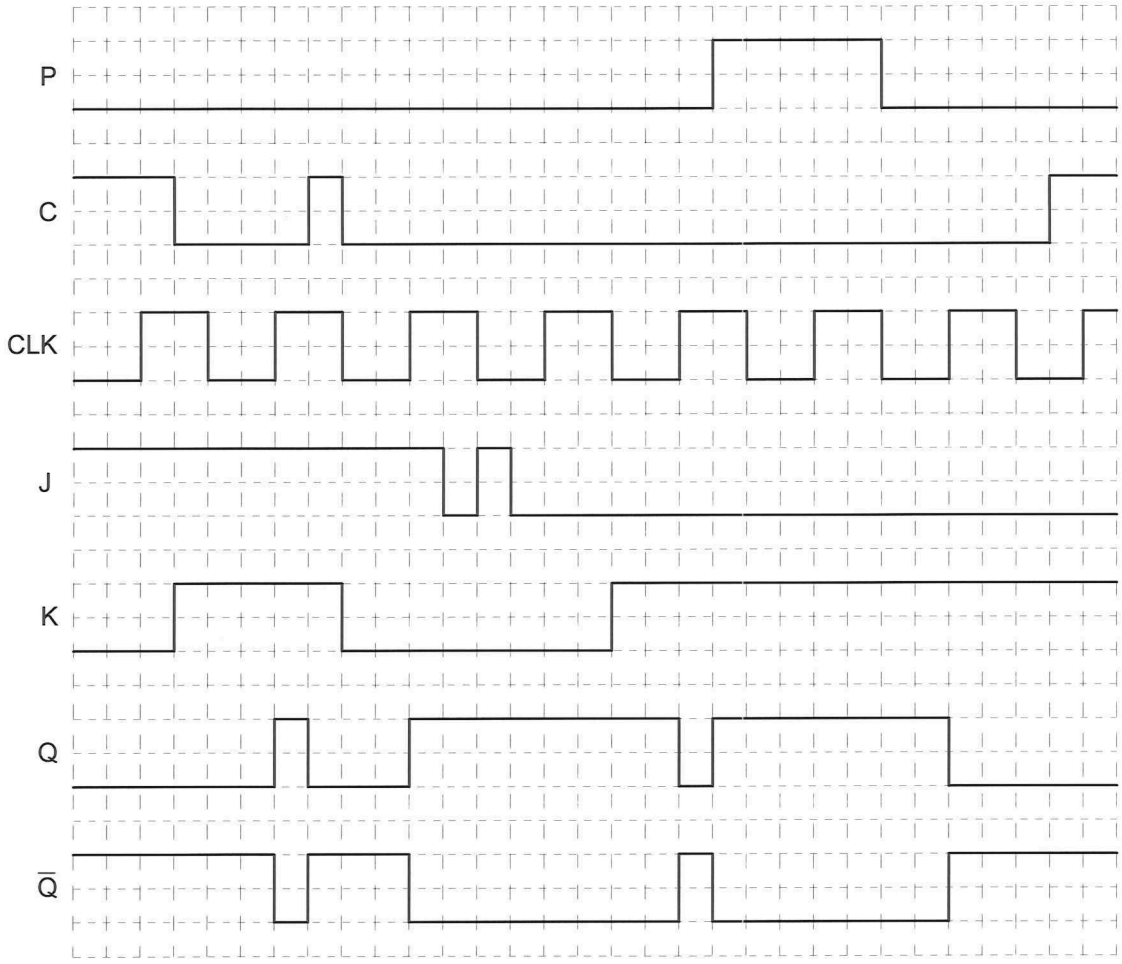


Figure 7

Practise drawing timing diagrams for J-K flip flops with preset and clear inputs by trying the following questions.

1. Complete the timing diagram below for the J-K FF shown in Figure 8.

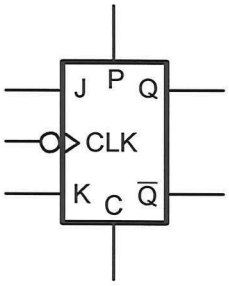
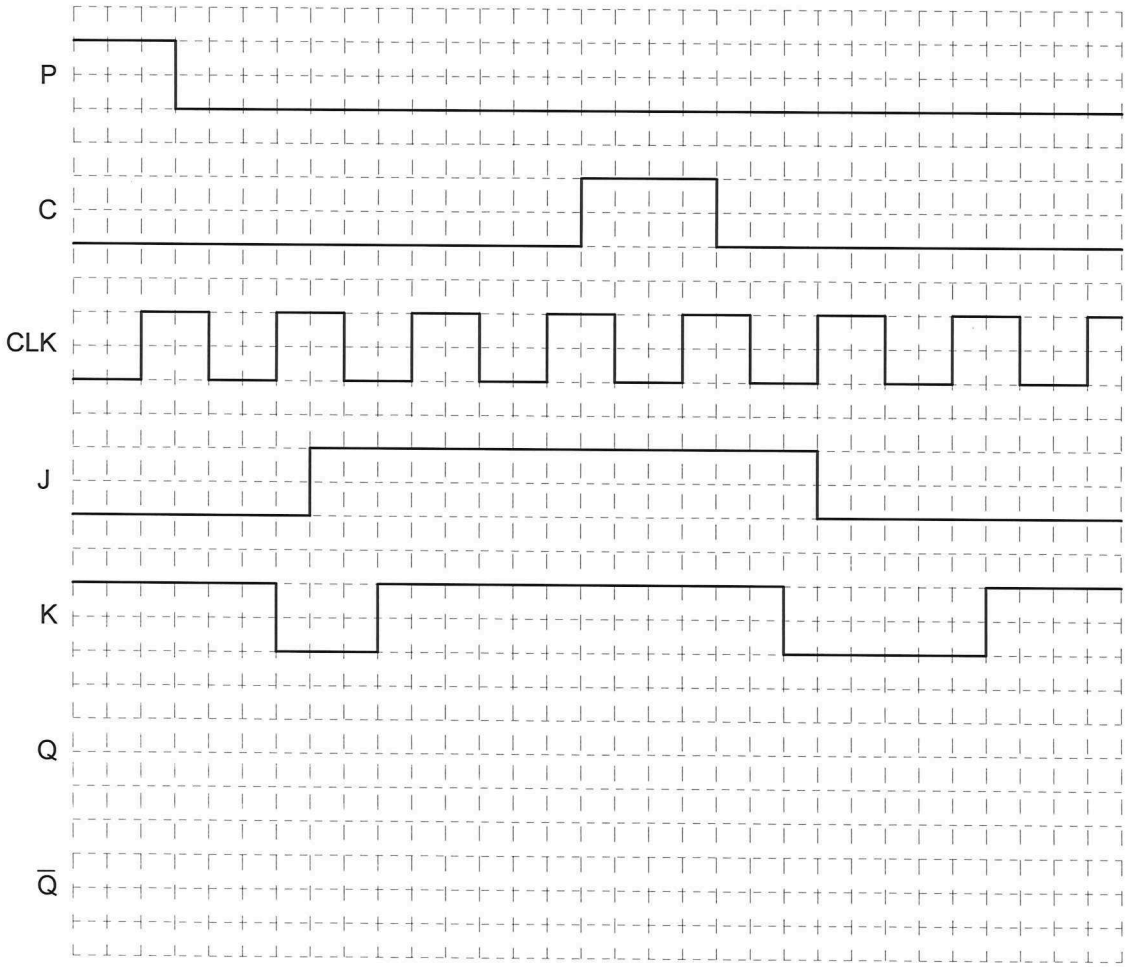


Figure 8



2. Complete the timing diagram below for the J-K FF shown in Figure 9.

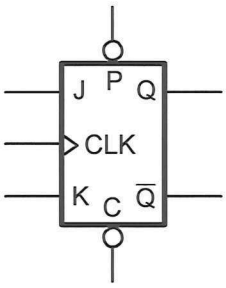
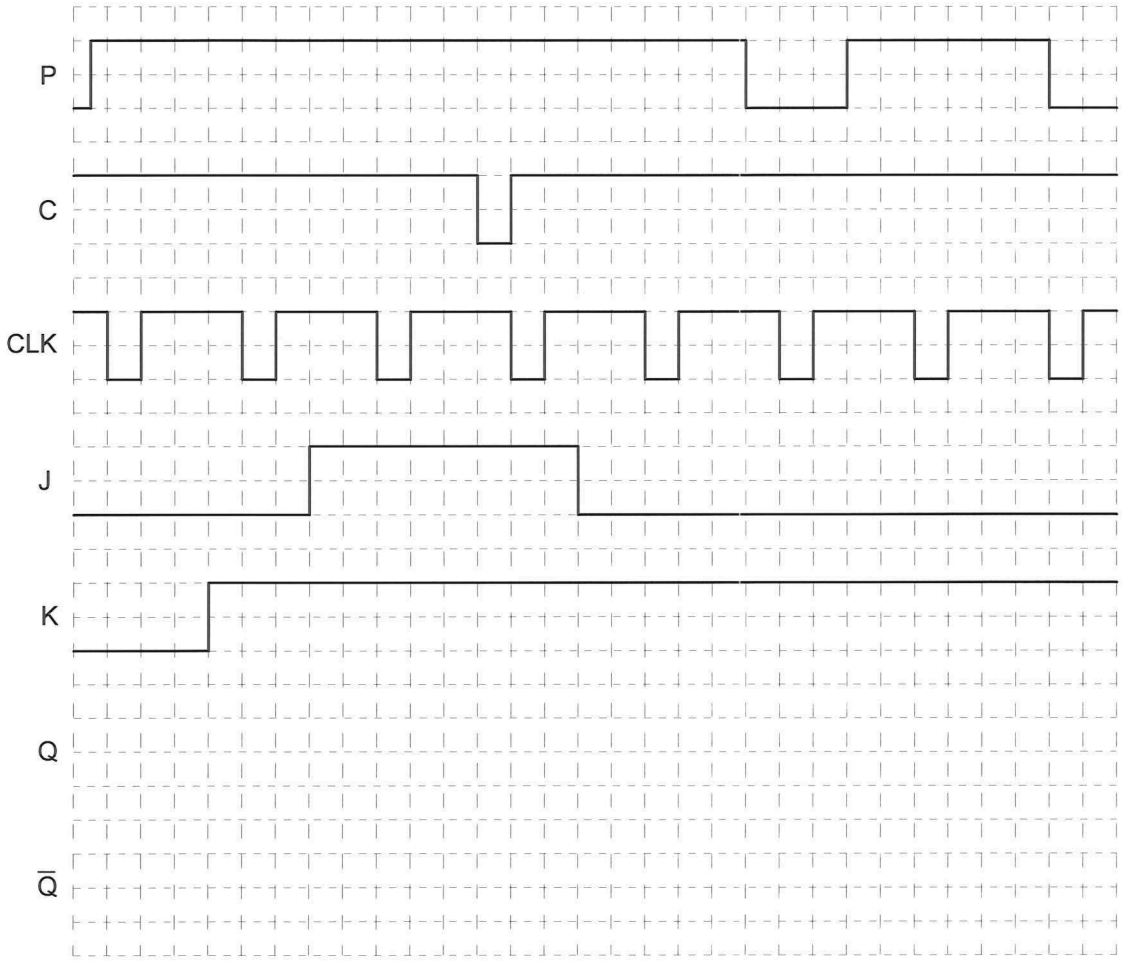


Figure 9



Applications for J-K flip flops

J-K flip flops with preset and clear inputs can be used instead of RS flip flops for all RS FF applications including the latch. This is because the *P* and *C* inputs of the J-K FF act in exactly the same way as the *S* and *R* inputs of the RS FF.

The J-K FF can be used to implement the circuit of a frequency divider by simply hard-wiring the *J* and *K* inputs to logic-1 (forcing the device to toggle each time it is locked) and applying the input signal to the flip flop's clock. This is shown in Figure 10 below using a positive edge-triggered J-K FF but a negative edge-triggered J-K FF could be used in exactly the same way.

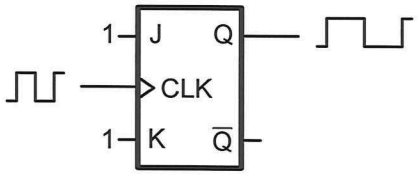


Figure 10 *A positive edge-triggered J-K FF used to implement the circuit of a frequency divider*

The J-K FF can also be used to implement a D-type FF as shown in Figure 11 below using a positive edge-triggered J-K FF (but again, a negative edge-triggered device would do the same job).

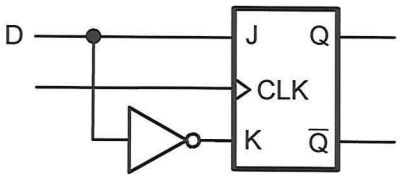


Figure 11 *A positive edge-triggered J-K FF used to implement the circuit of a D-type FF*

Like D-type flip flops, J-K flip flops are also used to make other important sequential logic circuits including *counters* and *shift registers*. As these devices are dealt with later in the workbook they're not discussed any further here.

Copyright: Not to be reproduced without permission

Student notes

Skill Practice 2

Practise testing and measuring sequential logic circuits (J-K flip flops)

This exercise is practise for the sorts of skills you may be required to perform in a practical test. Remember, in any practical tests you will be working alone so make sure that you can perform all the steps. It should take you about 45 minutes to complete this exercise.

Equipment

- Digital trainer
- logic probe
- 74LS76
- two BNC to alligator-clip leads
- hook-up wire (terminated)
- IC extractor

Remember:

Follow TAFE NSW WHS guidance at all times!

Work tasks

1. Read your WHS responsibilities at the top of the form below. Then conduct a WHS risk assessment and record your findings in the space provided.

Responsibilities of students under the Model WHS Act: s28

- Take reasonable care for your own health and safety by working safely at all times
- Take reasonable care to ensure that your acts or omissions don't put the health and safety of others at risk
- Follow all TAFE NSW WHS guidance and comply with all reasonable instructions from TAFE NSW staff to assist them in complying with the TAFE NSW WHS requirements
- In addition to the above, you must:
  - use and maintain machinery, tools and all other equipment properly and safely
  - ensure that your work area is free of hazards
  - notify a TAFE NSW staff member of actual or potential hazards
  - wear/use prescribed safety equipment
  - take notice of any safety signs and adhere to their instructions

Risks involved in this activity include:	**Control measures:**
■ Trip hazards (eg students bags) ■ Objects dropped on feet (while equipment is taken to and from benches).	■ Move bags and other objects from walkways ■ Plan lifting of equipment
Others: _____ _____ _____	Other: _____ _____ _____

My signature here indicates that I have read and understand my responsibilities under the Model WHS Act s28 (detailed above). I have also conducted a risk assessment before undertaking this activity and have identified measures to control these risks and have implemented them.

Signature: _____ Date: _____

2. Gather the equipment needed for this exercise.

3. Read the following points and make sure that you perform them when wiring all circuits:

   - Write the IC pin numbers of the devices you work with on the circuit that you're instructed to wire.
   - Handle all ICs carefully - remember ESD can damage them.
   - Ensure the power supply is connected to all ICs in the circuit.
   - When plugging hook-up wire into the breadboard to make connections to the IC's pins, use holes that are furthest away from the pin.

4. Wire the circuit of Figure 1 on the digital trainer. Pinout data for the 74LS76 is at the back of the book.

   Note: Use the *B* output of the P_B pulser button for the flip flop's clock input.

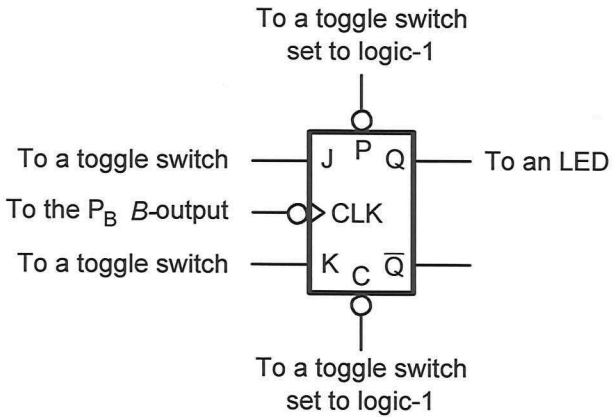


Figure 1

5. Turn on the trainer.

6. Use the C-input to reset the Q-output (if it's not already a logic-0).

 Copyright: Not to be reproduced without permission

7. Set the *J* and *K* inputs' toggle switches to logic-0.

8. **Momentarily** press the pulser button and note the logic level on output. Record your observation in Table 1 below.

9. Use the P-input to set the Q-output.

10. Repeat step 8.

11. Set the K-input's toggle switch set to a logic-1 and check that the J-input's toggle switch is still set to a logic-0.

12. Repeat step 8.

13. Leave the K-input's toggle switch set to a logic-1 and the J-input's toggle switch to a logic-0 and repeat step 8.

14. Set the J-input's toggle switch a logic-1 and the K-input's toggle switch to a logic-0.

15. Repeat step 8.

16. Leave the J-input's toggle switch set to a logic-1 and the K-input's toggle switch set to logic-0 and repeat step 8.

17. Set both the *J* and *K* inputs' toggle switches to logic-1.

18. Repeat step 8.

19. Leave the *J* and *K* inputs' toggle switches set to a logic-1 and repeat step 8.

Table 1

J	K	Q_n	Q_{n+1}
0	0	0	
0	0	1	
0	1	1	
0	1	0	
1	0	0	
1	0	1	
1	1	1	
1	1	0	

Question 1
What happens to the Q-output when *J* and *K* inputs are logic-0 and the device is clocked?

Question 2
What happens to the Q-output when the K-input is logic-1, the J-input is logic-0 and the device is clocked?

Question 3
What happens to the Q-output when the J-input is logic-1, the K-input is logic-0 and the device is clocked?

Question 4
What happens to the Q-output when the *J* and *K* are logic-1 and the device is clocked?

The teacher needs to check your work at this point...

20. Connect the J-K flip flop's $\overline{Q}$-output to an LED.

21. Set the *Range* control of the trainer's in-built function generator to the ×*10* position.

22. Turn the trainer's *Frequency* control to the "1" setting.

23. Disconnect the J-K flip flop's clock input from the toggle switch and connect it to the trainer's ***TTL Mode*** output instead.

24. Leave the J and K inputs' toggle switches set a logic-1.

Self Check: If steps 21 to 23 have been done correctly, the LED connected to the J-K flip flop's outputs should be flashing alternately. If not, check that you have followed the instructions correctly. If you still can't get the outputs to flash call the teacher.

25. Set the trainer's *Range* control to the ×*100* position.

Question 6
Why do the LEDs now look as though they're not flashing anymore?

_____

_____

The teacher needs to check your work at this point...

26. Set the *Range* control to the *×10k* position.

27. Set up the bench-mounted CRO to display both channels. Adjust the *Horizontal Timebase* control to the *20µs/div* setting and the *Vertical Timebase* control to the *1V/div* setting.

28. Connect the CRO's Channel 1 input to the J-K flip flop's clock input.

29. Connect the CRO's Channel 2 input to the J-K flip flop's Q-output.

30. Measure the clock signal's period (on Channel 1) as accurately as possible. Record your measurement in Table 2 below.

    Note: If available, use the CRO's measurement tools to perform this step.

31. Use your measurement to calculate the clock signal's frequency. Record your calculation in Table 2 also.

    Note: Again, you can use the CRO's measurement tools for this purpose instead of calculating it.

32. Repeat Steps 30 and 31 for the output signal (on Channel 2).

Table 2

Clock signal		Output signal	
Period	**Frequency**	**Period**	**Frequency**

Question 7
What is the relationship between the frequency of the two signals?

_____

_____

Question 8
What is the name of the circuit that the J-K FF has been wired to operate as?

_____

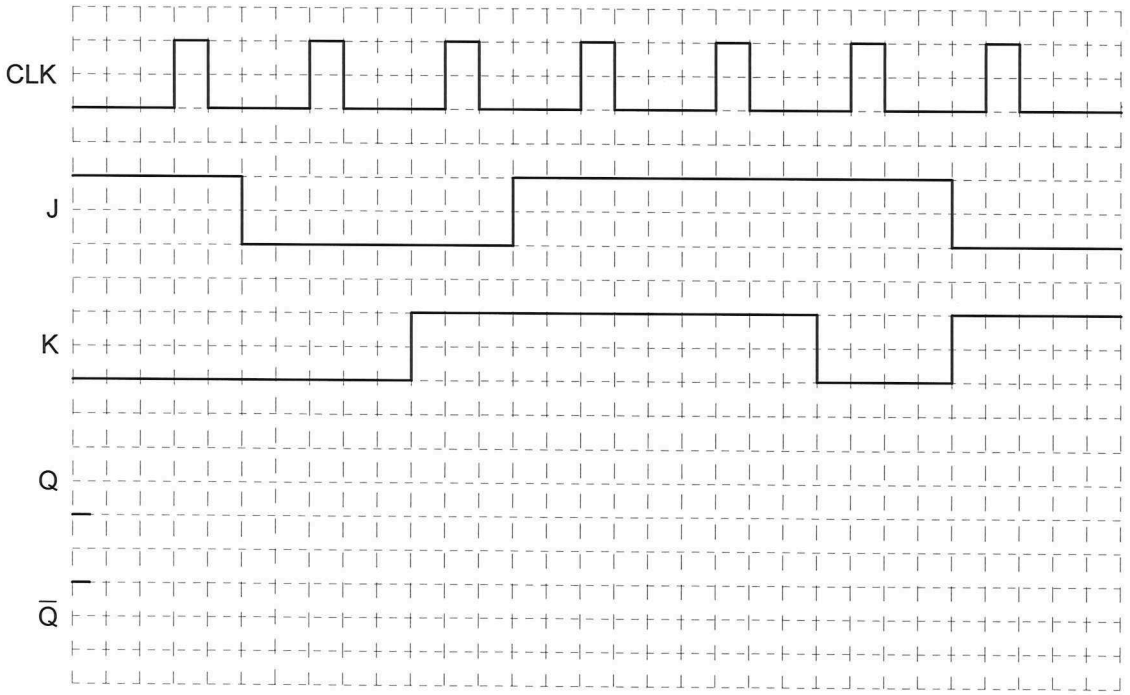
Copyright: Not to be reproduced without permission

The teacher needs to check your work at this point...

Review Questions

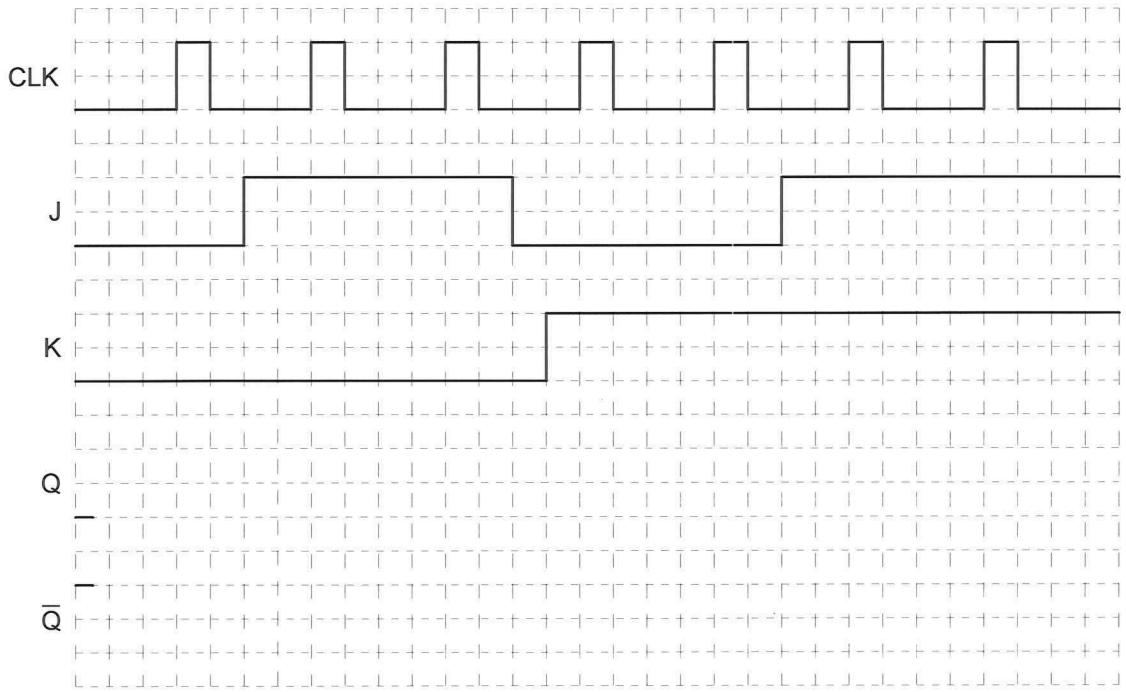
Answer these questions to check your understanding of what you have learnt from this chapter. Doing this will also help to prepare you for the assessments.

1. Complete the timing diagram below for a negative edge-triggered J-K FF without preset and clear inputs. Assume the Q-output starts as logic-0.



Copyright: Not to be reproduced without permission

2. Complete the timing diagram below for a positive edge-triggered J-K FF without preset and clear inputs. Assume the Q-output starts as logic-0.



Questions 3 to 7 refer to the device in Figure 1 below.

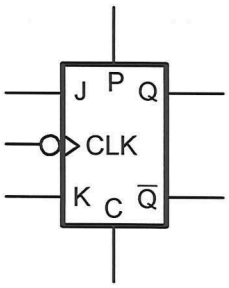


Figure 1

3. Which input or inputs are asynchronous?

- ☐ *J* and *K* only
- ☐ CLK only
- ☐ *P* and *C* only
- ☐ All of them
- ☐ None of them

4. Which input or inputs are synchronous?

- ☐ *J* and *K* only
- ☐ CLK only
- ☐ *P* and *C* only
- ☐ All of them
- ☐ None of them

5. Which input or inputs are negative edge-triggered?

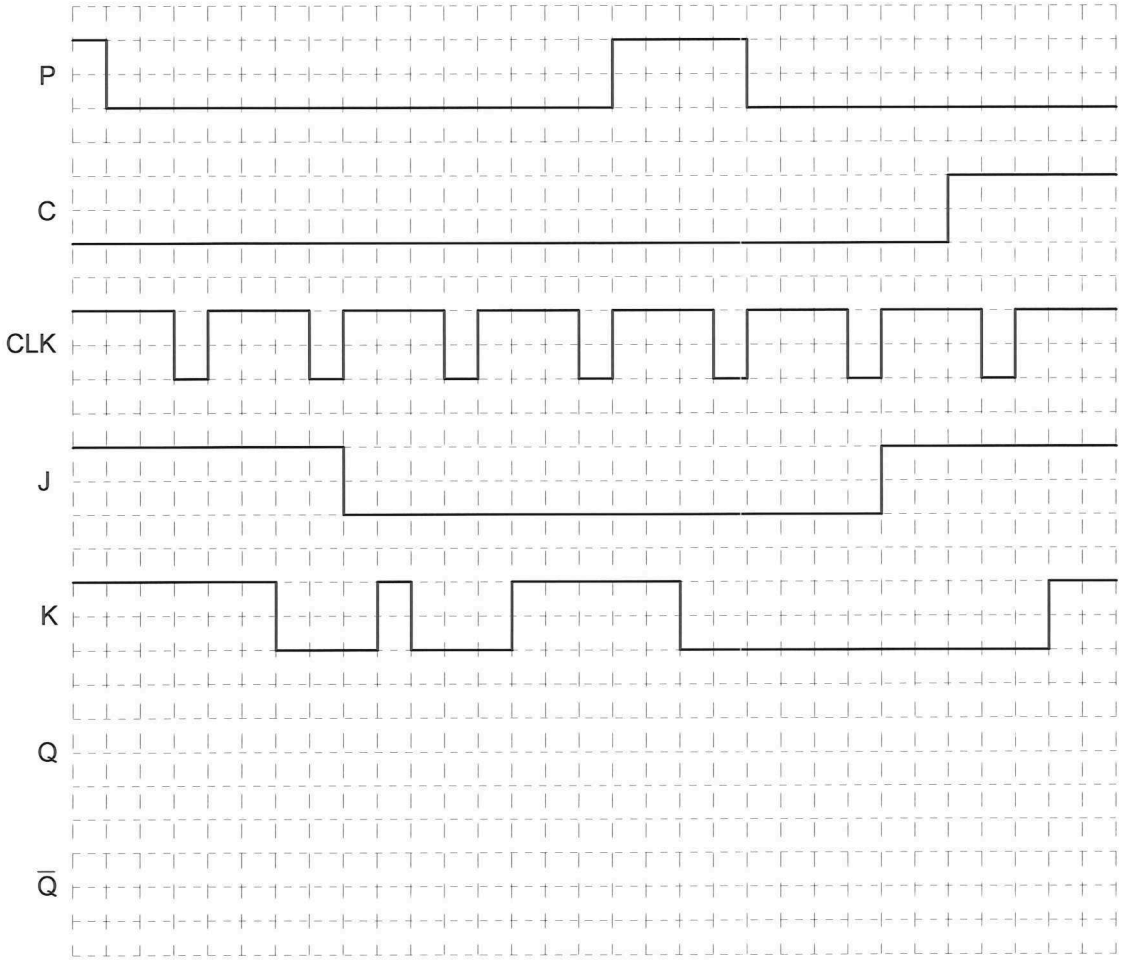
- ☐ *J* and *K* only
- ☐ CLK only
- ☐ *P* and *C* only
- ☐ All of them
- ☐ None of them

 Copyright: Not to be reproduced without permission

6. Which inputs are active low?

- ☐ *J* and *K* only
- ☐ CLK only
- ☐ *P* and *C* only
- ☐ All of them
- ☐ None of them

7. Complete the timing diagram below for the device in Figure 1.



8. Complete the timing diagram below for the device in Figure 2.

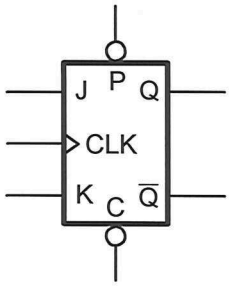
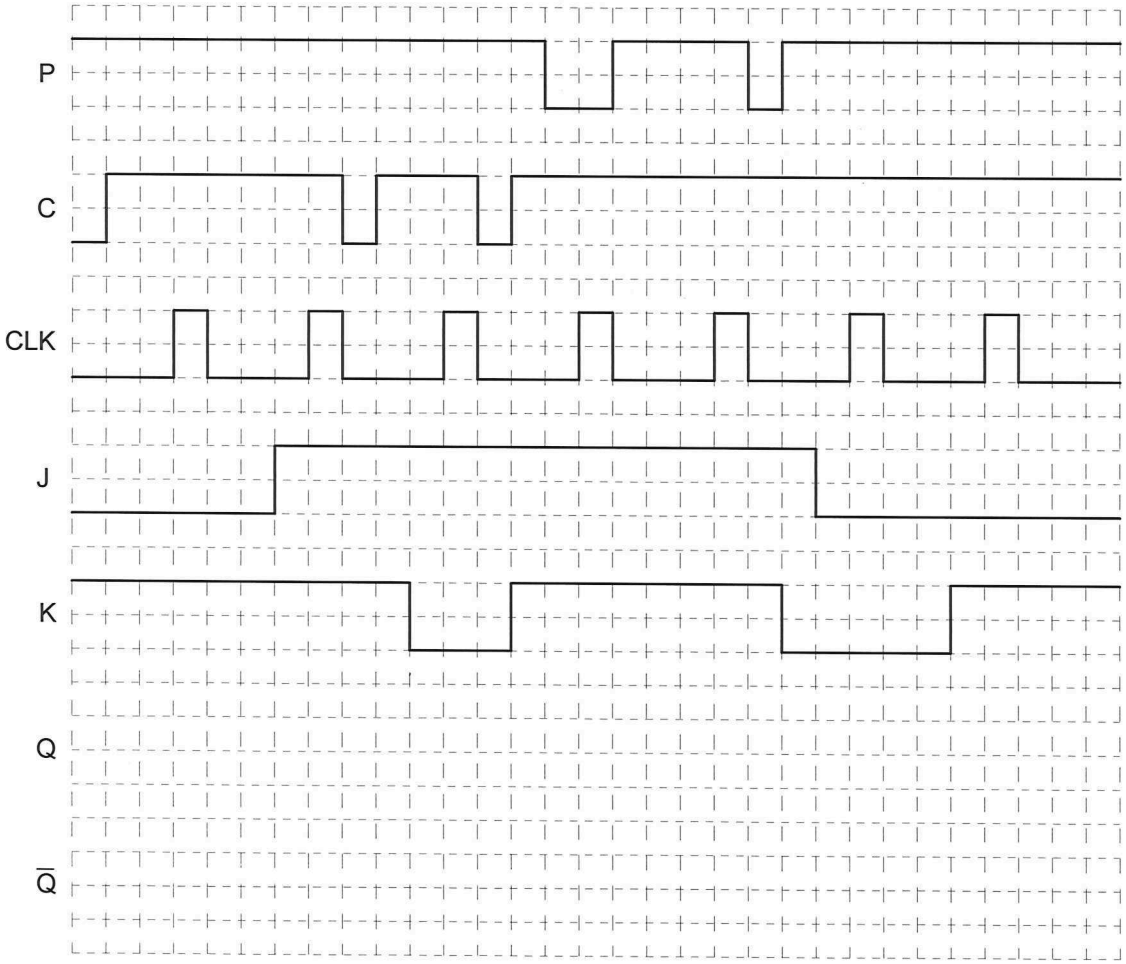


Figure 2



9. Unused active low *P* and *C* inputs of D-type and J-K flip flops should be

- ☐ left not connected to anything.
- ☐ hard-wired to logic-0
- ☐ hard-wired to logic-1.
- ☐ tied to the supply rail using a pull-up resistor.

10. What problem can be caused by leaving the unused preset and clear inputs of D-type and J-K inputs not connected to anything?

_____

_____

_____

11. List four applications for J-K flip flops.

_____

_____

_____

_____

12. Draw the circuit that implements a frequency divider using a J-K FF.

13. Draw the circuit that implements a D-type FF using a J-K FF.

Questions 14 to 19 refer to the 7473, 7476 and 4027 ICs. Data for these devices can be found on the internet at various websites including https://au.element14.com.

14. Which of the three devices does not have a preset input?

_____

15. Which of the three devices has a positive edge-triggered clock?

_____

16. Which of the three devices has active low clear inputs?

_____

17. Which of the three devices has a active high preset input?

_____

18. Which of the three devices contains flip flops that **don't** have a $\overline{Q}$-output?

_____

19. Are the pin assignments the same for the three devices?

_____

Copyright: Not to be reproduced without permission

Student notes

Section 3 Multivibrators

Purpose To develop your understanding of multivibrators and to develop your ability to interpret the manufacturers' data sheets for common commercially available monostable multivibrator ICs.

Objectives At the end of this section you should be able to:

- Explain the basic principle of operation of a *multivibrator* (at the block diagram level only)
- Explain the difference between a *bistable*, *astable* and *monostable* multivibrators and given applications for each
- State the commonly used alternate name for monostable multivibrators
- Specify which two components are most widely to be used to set the duration of a monostable multivibrator's unstable state
- Draw and/or interpret the timing diagram of monostable multivibrators
- Explain the difference between *retriggerable* and *non-retriggerable* monostable multivibrators
- Interpret manufacturers' data for common commercially available monostable multivibrator ICs
- Use a logic probe to test and measure monostable multivibrator digital ICs

Introduction

The last couple of chapters lessons have concentrated on explaining the operation of various flip flops. What has not been mentioned is that flip flops belong to a class of devices called *multivibrators*. A multivibrator is a sequential logic circuit or device with an output that simply switches between one of two logic levels (either continuously or on demand).

There are three basic types of multivibrator:

- the bistable multivibrator
- the astable multivibrator
- the monostable multivibrator

The bistable multivibrator is a device that switches from one logic level to another on demand from an external trigger source and examples include: RS FFs, D-type FFs and J-K FFs. This chapter takes a brief look at the principles of operation of astable multivibrators then takes a close look at monostable multivibrators and some dedicated monostable multivibrator ICs.

Astable multivibrators

The letter "a" when used as a prefix in English means *not* or *without*. For example, a person who is said to be *apolitical* doesn't have an interest in politics, the political system or political parties (and probably only votes because its compulsory to do so in Australia). That being the case, an astable multivibrator is a device or circuit that has an output without a default logic level. Instead, the output continuously toggles between logic-0 and logic-1 (usually at a fixed rate). In other words, astable multivibrators are circuits that output a pulse train. For this reason they're often also called *digital oscillators*.

The block diagram of an astable multivibrator is shown in Figure 1 below. Notice that it doesn't have any inputs. This means that astable multivibrators don't need to be triggered (like flip flops do) and are said to be *free-running*. [Note: Like nearly all other circuits, the astable multivibrator must be powered by a DC source and so it will have power supply inputs but these are not usually shown on block diagrams.]

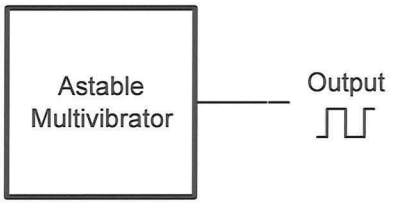


Figure 1 *Block diagram of an astable multivibrator*

Applications
You'll learn much more about oscillators of all types including the astable multivibrator in a later unit. At this stage, you're only required to know that they're used for generating digital clock signals. All microprocessor-based systems require a clock signal and most other digital systems do too so astable multivibrators are used extensively.

Monostable multivibrators

As you probably know, "mono" means *one*. That being the case, although multivibrators have an output that can switch between two logic levels, the output of a monostable multivibrator always defaults to one of them. When the device is triggered, its output switches to the alternate logic level temporarily (and usually for a set duration) and automatically returns to the default level.

The trigger input to monostable multivibrators is just called the *trigger* (instead of clock) and is edge-sensitive. It can be either positive or negative edge-triggered and this is denoted in the standard way. Like flip flops, monostable multivibrators usually have a complementary output (that is, a $\overline{Q}$ output). Figure 2 below shows the block diagram of a positive edge-triggered monostable multivibrator.

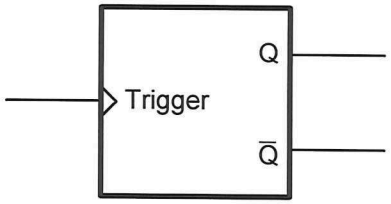


Figure 2 *A positive edge-triggered monostable multivibrator*

The operation of a monostable multivibrator can be described using a timing diagram. Figure 3 below shows the logic levels on the outputs of the device in Figure 2 both before and after the device is triggered.

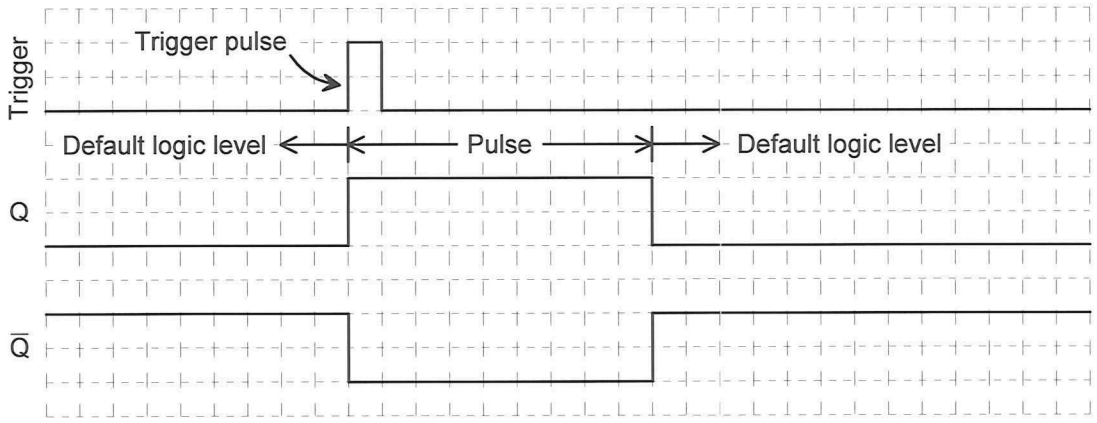


Figure 3

When the operation of the monostable multivibrator is shown in this way, we can see why it is sometimes referred to as a *oneshot*.

If the device in Figure 2 had been negative edge-triggered then the timing diagram would look like that in Figure 4 below.

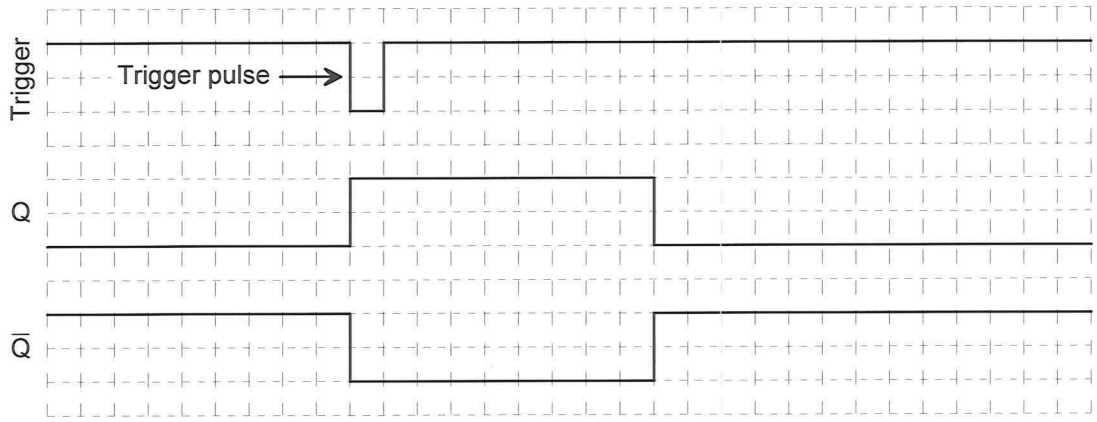


Figure 4

The duration of the device's unstable state (the *pulse*) is usually set by an external RC circuit. In this way, the pulse-width can be set to any length of time (within the limits of the device's specifications). Some monostable multivibrator ICs provide an in-built resistor (typically 10kΩ) so that only an external capacitor is needed to set the pulse-width.

If a monostable multivibrator is used in an application where the pulse-width is required to be changeable, either a variable resistor or variable capacitor can be used in the external RC network.

Applications

Monostable multivibrators can be used wherever an event is required to occur for only a predetermined amount of time. Obvious everyday examples of where they could be used includes any of the many access control systems that permit people to enter buildings, a train stations, and so.

For example, when a person brings their access fob close to the wall-mounted reader, the door is unlocked for just enough time for the person to be able to open the door. You've probably heared that characteristic click from inside the door-strike, a solenoid has been energised.

But, as the person is only given a temporary opportunity to open the door, the solenoid must only be energised temporarily. The solenoid then can be controlled by a monostable multivibrator (via an interface circuit).

Another application for the monostable multivibrator is *pulse stretching*. Pulse stretching is used in all sorts of situations. For example, the buzz from a shop's door alarm may not attract a shop assistant's attention if the person entering the shop breaks the beam only very briefly. So, instead of the detector circuit driving the buzzer directly, it can be used to trigger a monostable multivibrator with a two second pulse. This way, no matter how briefly the customer breaks the beam, the buzzer will sound for two full seconds.

Another circuit that uses a pulse stretcher is the logic probe. Sometimes, pulses are too quick for the *HI* and *LO* LEDs to react to and display. To remedy this, logic probes usually have a third LED (a yellow one) which is connected to the output of a 0.5 second pulse stretcher. Any pulse on the input, no matter how quick, always triggers this circuit causing the yellow LED to light up.

Non-retriggerable monostable multivibrators (discussed next) can be used for switch-debouncing (instead of the method described in the Part 1 workbook). Where this is done, the oneshot is used as a type of buffer between the switch and the circuit that the switch's data connects to. Switch-bounce typically lasts about 10ms so the duration of the oneshot's pulse is made a little longer to avoid the multiple transitions from generating more than one pulse.

Non-retriggerable and retriggerable monostable multivibrators

An important difference between monostable multivibrators relates to how they react if triggered when the output is already in its unstable state (due to the device having been triggered previously). Non-retriggerable monostable multivibrators ignore trigger signals while the device's output is in its unstable state. To elaborate, suppose a non-retriggerable oneshot has been programmed to have a ten second pulse. Then, suppose this device is triggered once then triggered a second time five seconds later. The second pulse has no effect on the device and its output stays in its unstable state for only ten seconds. This is shown in Figure 5 below.

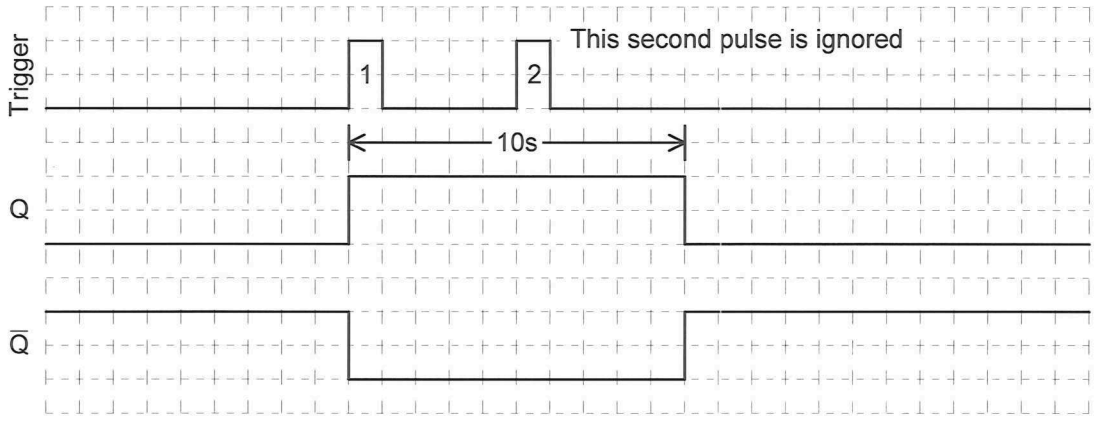


Figure 5

Retriggerable multivibrators don't ignore trigger signals while the device's output is in its unstable state. Instead, any trigger signal during this time causes the pulse's duration to restart. To use an example similar to the one above, if a 10s retriggerable oneshot is triggered once then triggered a second time five seconds later, the device's output stays in its unstable state for fifteen seconds. This is shown in Figure 6 below.

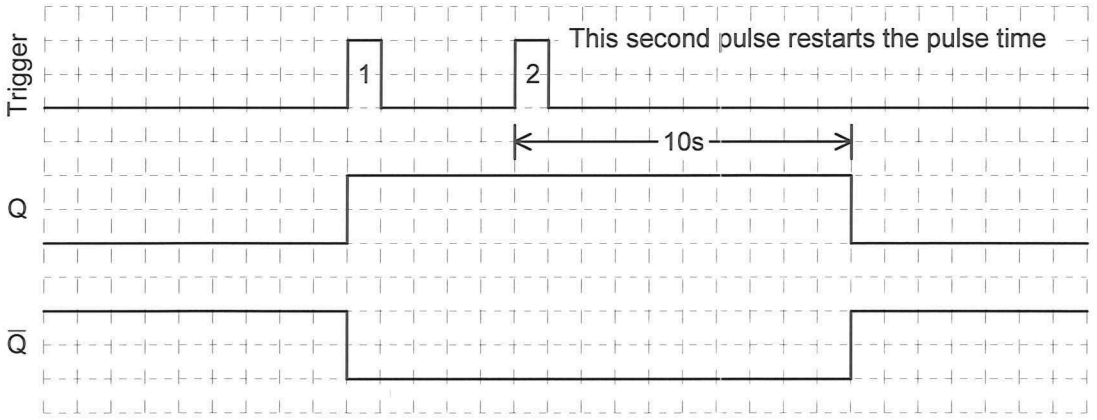


Figure 6

If trigger pulses occur regularly enough, a monostable multivibrator's output can be kept in its unstable state indefinitely. This is illustrated in Figure 7 below.

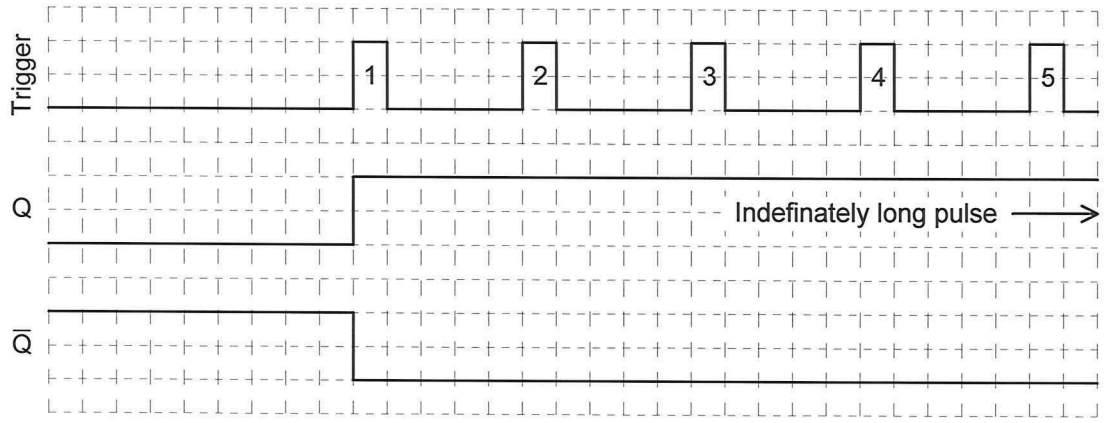


Figure 7

Monostable multivibrator ICs

Monostable multivibrators are important devices in the operation of digital electronics equipment and so there are TTL and CMOS ICs dedicated to the function. These include the 74122, 74123, 74221, 4047, 4098 and the 4528. To give you a feel for working with monostable multivibrator ICs we'll look at just one of these devices and discuss its operation in detail.

The 74221 monostable multivibrator

This TTL device is a dual non-retriggerable monostable multivibrator (that is, there are two oneshots inside the IC) both having a clear input, complementary outputs and two triggers inputs: one positive edge-triggered and the other negative. The pinout for the 74221 is shown in Figure 8 below.

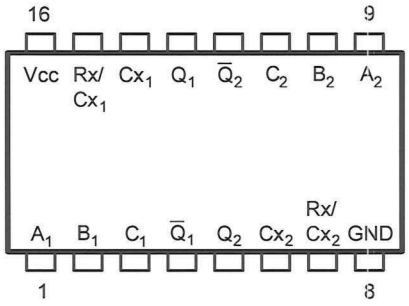


Figure 8 *Pinout data for the 74221*

To help make the device a little easier to understand, it has been redrawn in Figure 9 below in block diagram form with the pins arranged more conveniently. As there are two oneshots inside the IC there are two blocks in Figure 9.

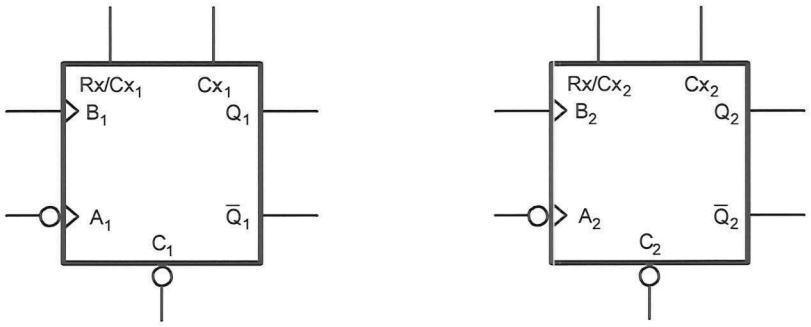


Figure 9 *Block diagrams for the two devices in the 74221 (power supply connections not shown)*

As the block diagrams show, these devices have an active low clear input (C_1 & C_2). According to the manufacturer's data sheet, the clear input is asynchronous. Also, when asserted, the clear input overrides the other inputs.

As already mentioned above, each device has two trigger inputs - A and B. The block diagrams show that input A is the negative edge-triggered trigger input and B is the positive edge-triggered input.

The C_X and R_X/C_X inputs are used to connect the external timing components (that is, the resistor and capacitor) to the device. The components' values determine the output pulse's duration.

Figure 10 below shows the minimum wiring requirements needed to operate either one of the devices inside the 74221 as a positive edge-triggered monostable multivibrator.

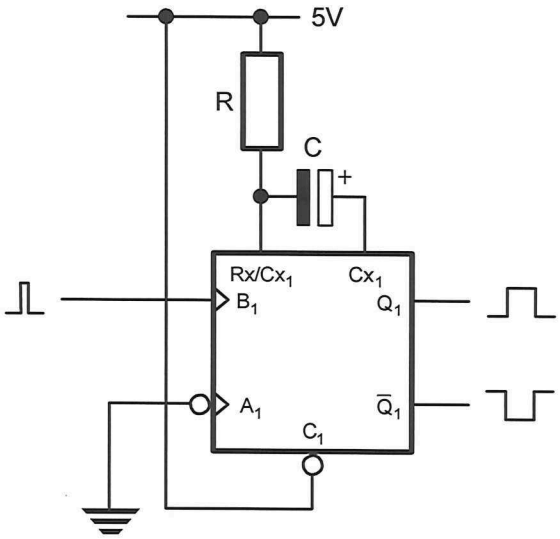


Figure 10 *The minimum wiring requirements to operate either of the devices inside the 74221 as a positive edge-triggered monostable multivibrator*

Figure 11 below shows the minimum wiring requirements needed to operate either one of the devices inside the 74221 as a negative edge-triggered monostable multivibrator.

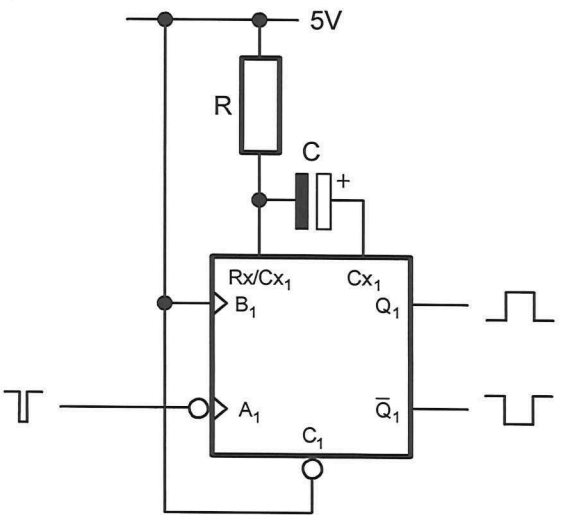


Figure 11 *The minimum wiring requirements to operate either of the devices inside the 74221 as a negative edge-triggered monostable multivibrator*

As already mentioned, the pulse's duration is set by an externally connected resistor and capacitor and the equation below is given by the manufacturers of the 74221 to calculate its pulse-width. It should be pointed out though that the pulse-width equation varies from one IC to another.

$$t_w = 0.7 \times R_X \times C_X$$

Where:
t_w is the pulse width (in seconds)
R_X is the value of the external resistor
C_X is the value of the external capacitor

This equation assumes that the capacitor used is equal or greater than 1nF. A different equation must be used when capacitor values below 1nF are used.

Practise calculating the pulse width of a monostable multivibrator for yourself by attempting the question below.

1. Calculate the duration of the pulse for the device shown in Figure 12 below.

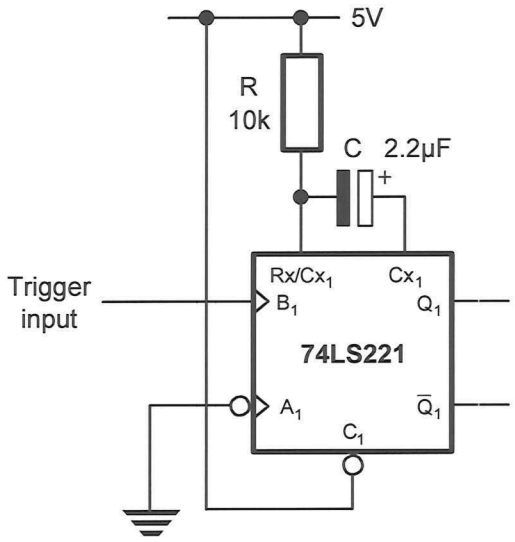


Figure 12

Skill practice 3

Practise testing and measuring monostable multivibrator digital ICs

This exercise is practise for the sorts of skills you may be required to perform in a practical test. Remember, in any practical tests you will be working alone so make sure that you can perform all the steps. It should take you about 1 hour to complete this exercise.

Equipment

- Digital trainer
- logic probe
- 74LS221
- 100kΩ ¼W resistor
- 470kΩ ¼W resistor
- 10µF 25V electrolytic capacitor
- hook-up wire (terminated)
- IC extractor

Remember:

Follow TAFE NSW WHS guidance at all times!

Work tasks

1. Read your WHS responsibilities at the top of the form below. Then conduct a WHS risk assessment and record your findings in the space provided.

Responsibilities of students under the Model WHS Act: s28

- Take reasonable care for your own health and safety by working safely at all times
- Take reasonable care to ensure that your acts or omissions don't put the health and safety of others at risk
- Follow all TAFE NSW WHS guidance and comply with all reasonable instructions from TAFE NSW staff to assist them in complying with the TAFE NSW WHS requirements
- In addition to the above, you must:
  - use and maintain machinery, tools and all other equipment properly and safely
  - ensure that your work area is free of hazards
  - notify a TAFE NSW staff member of actual or potential hazards
  - wear/use prescribed safety equipment
  - take notice of any safety signs and adhere to their instructions

Risks involved in this activity include:	**Control measures:**
■ Trip hazards (eg students bags) ■ Objects dropped on feet (while equipment is taken to and from benches).	■ Move bags and other objects from walkways ■ Plan lifting of equipment
Others: _____ _____ _____	Other: _____ _____ _____

My signature here indicates that I have read and understand my responsibilities under the Model WHS Act s28 (detailed above). I have also conducted a risk assessment before undertaking this activity and have identified measures to control these risks and have implemented them.

Signature: _____ Date: _____

2. Gather the equipment needed for this exercise.

3. Read the following points and make sure that you perform them when wiring all circuits:

   - Write the IC pin numbers of the devices you work with on the circuit that you're instructed to wire.
   - Handle all ICs carefully - remember ESD can damage them.
   - Ensure the power supply is connected to all ICs in the circuit.
   - When plugging hook-up wire into the breadboard to make connections to the IC's pins, plug the hook-up wire into the furthest free hole away from the pin.

4. Wire the circuit of Figure 1 on the digital trainer ensuring that you connect the power supply to the IC. Pinout data for the 74LS221 can be found in Appendix 6.

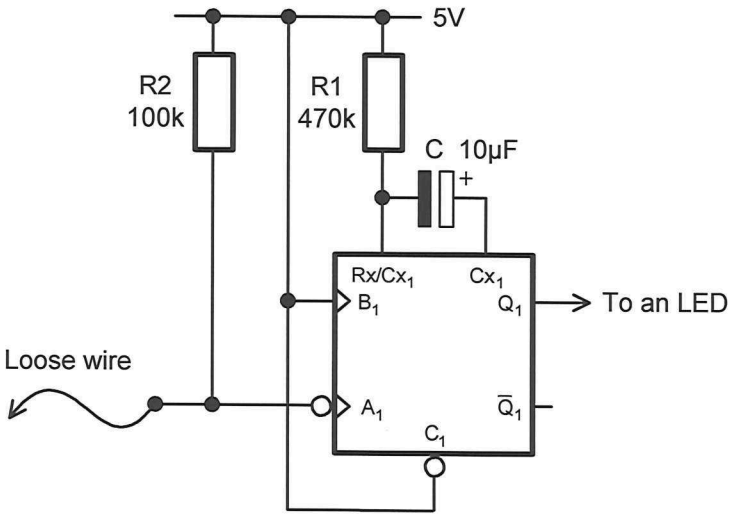


Figure 1

 Copyright: Not to be reproduced without permission

5. Turn on the trainer.

   Self check: The Q-output should be logic-0. **Momentarily** touch the end of the floating wire to ground. When you do this it triggers the circuit and the Q-output should change to a logic-1 for a short time then return to a logic-0. If your circuit doesn't do this then turn off the trainer and check your wiring. If it still doesn't work, call the teacher.

   Question 1
   What is this circuit?

   _____

6. Trigger the circuit again and measure the duration of the output pulse. Record your measurement in Table 1 below.

7. Calculate and record the duration of the pulse using the circuit's component values and the equation on page 3-10.

Table 1

Measured pulse width	Calculated pulse width

Question 2
Are the measured and calculated pulse-widths the same? Why might there be differences?

_____

_____

Question 3
Is the circuit positive edge-triggered or negative edge-triggered? Explain why connecting the trigger input to ground to trigger the device proves this.

_____

_____

_____

8. Trigger the circuit again and time the pulse.

9. While the device's output is in its unstable state trigger it several more times.

Question 4
Did re-triggering the device while its output was in its unstable state extend the pulse's duration?

Question 5
What does this tell us about this monostable multivibrator?

The teacher needs to check your work at this point...

Copyright: Not to be reproduced without permission

Student notes

Review questions

Answer these questions to check your understanding of what you have learnt from this chapter. Doing this will also help to prepare you for the assessments.

1. A multivibrator's output switches between

   ☐ two logic levels.
   ☐ three logic levels.
   ☐ an infinite number of logic levels.

2. A multivibrator's output switches

   ☐ continuously.
   ☐ only on demand.
   ☐ either continuously or on demand depending on the device.

3. A bistable multivibrator's output has

   ☐ no stable states.
   ☐ one stable state.
   ☐ two stable states.
   ☐ an infinite number of stable states.

4. Which device or devices are examples of a bistable multivibrator?

   ☐ A squarewave oscillator
   ☐ A NAND RS FF
   ☐ A NOR RS FF
   ☐ A J-K FF
   ☐ A oneshot

5. An astable multivibrator's output has

   ☐ no stable states.
   ☐ one stable state.
   ☐ two stable states.
   ☐ an infinite number of stable states.

 Copyright: Not to be reproduced without permission

6. Which device or devices are examples of an astable multivibrator?

- ☐ A squarewave oscillator
- ☐ A NAND RS FF
- ☐ A NOR RS FF
- ☐ A J-K FF
- ☐ A oneshot

7. A monostable multivibrator's output has

- ☐ no stable states.
- ☐ one stable state.
- ☐ two stable states.
- ☐ an infinite number of stable states.

8. Which device or devices are examples of a monostable multivibrator?

- ☐ A squarewave oscillator
- ☐ A NAND RS FF
- ☐ A NOR RS FF
- ☐ A J-K FF
- ☐ A oneshot

9. Which of the multivibrators doesn't need a trigger input or inputs?

- ☐ Bistable
- ☐ Astable
- ☐ Monostable

10. Which of the multivibrators needs a trigger signal to change from a logic-0 to a logic-1 but not to change back to a logic-0 from a logic-1?

- ☐ Bistable
- ☐ Astable
- ☐ Monostable

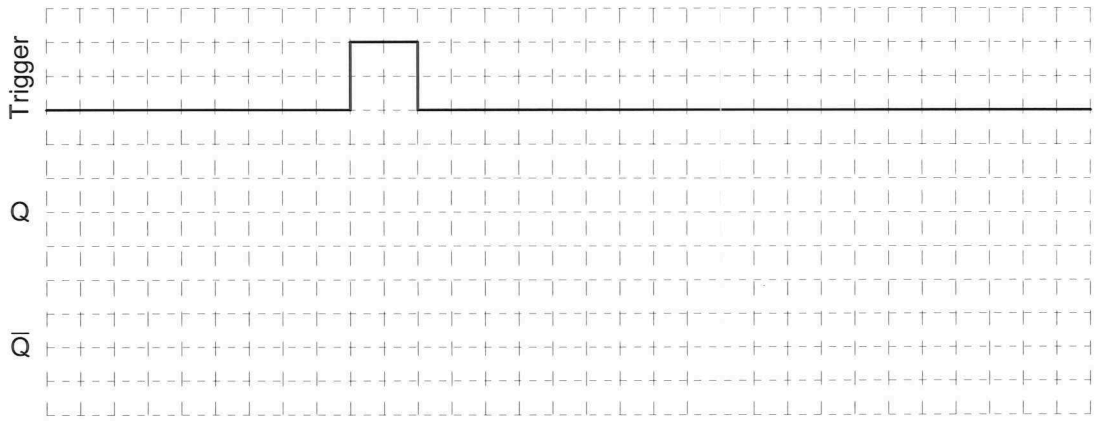
11. Draw the block diagram of an astable multivibrator.

12. Give an application for an astable multivibrator.

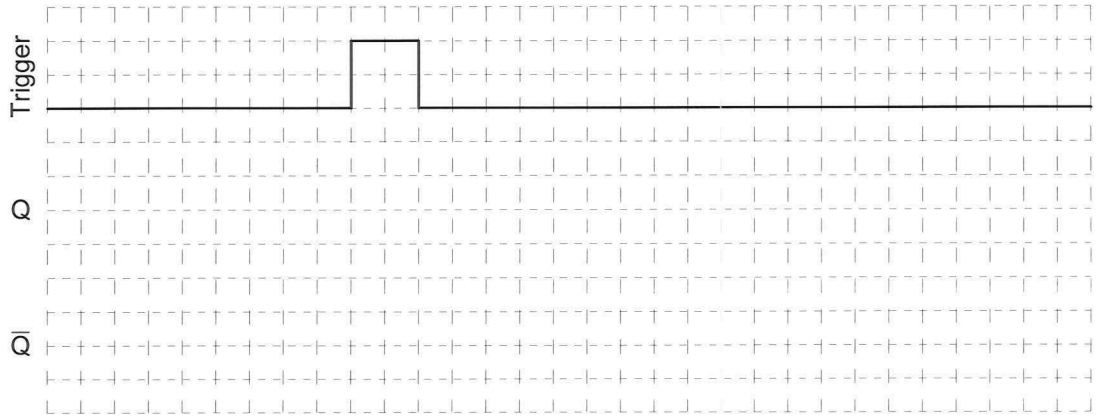
13. Draw the block diagram of a negative edge-triggered monostable multivibrator.

14. Give two applications for a monostable multivibrator.

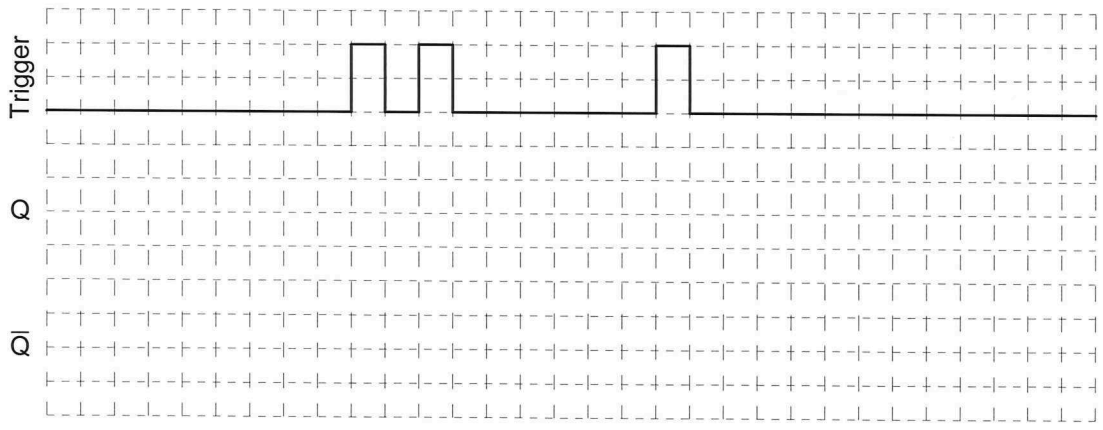
15. On the timing diagram below, show what would happen to the Q and $\overline{Q}$ outputs of a positive edge-triggered monostable multivibrator for the trigger signal shown. Assume the output pulse width is 8 divisions.



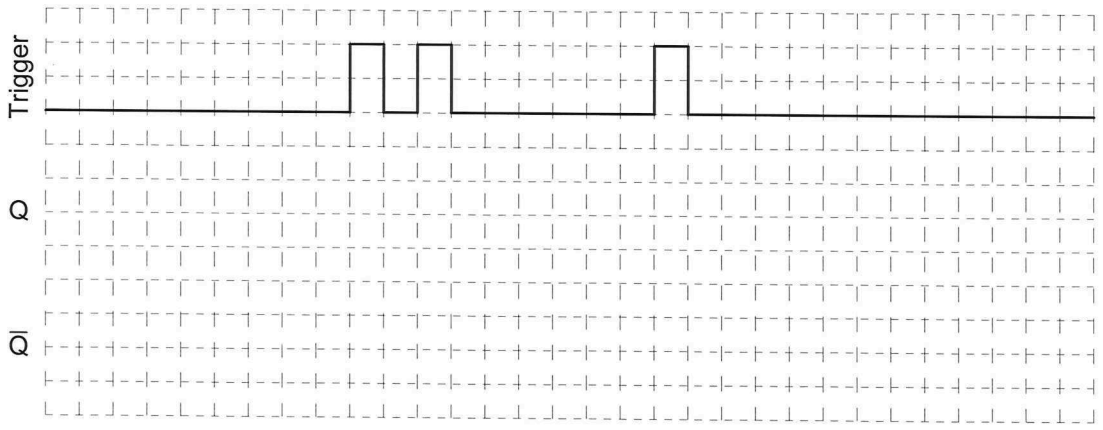
16. On the timing diagram below, show what would happen to the Q and $\overline{Q}$ outputs of a negative edge-triggered monostable multivibrator for the trigger signal shown. Assume the output pulse width is 8 divisions.



17. On the timing diagram below, show what would happen to the Q and $\overline{Q}$ outputs of a positive edge-triggered non-retriggerable monostable multivibrator for the trigger signal shown. Assume the output pulse width is 8 divisions.



18. On the timing diagram below, show what would happen to the Q and $\overline{Q}$ outputs of a positive edge-triggered retriggerable monostable multivibrator for the trigger signal shown. Assume the output pulse width is nominally 8 divisions.



 Copyright: Not to be reproduced without permission

Questions 19 to 26 refer to the 74LS122 IC. Data for the device can be found in Appendix 6 though you may need to obtain its full data sheet to answer some of the questions

19. The device is

- ☐ retriggerable.
- ☐ non-retriggerable.

20. The device has

- ☐ one trigger input that is positive edge-triggered.
- ☐ one trigger input that is negative edge-triggered.
- ☐ two trigger inputs - one positive and one negative edge-triggered.

21. The clear input is

- ☐ active low.
- ☐ active high.
- ☐ negative edge-triggered.
- ☐ positive edge-triggered.

22. What are the minimum and maximum values that can be used for the externally connected resistor?

_____

_____

23. What are the minimum and maximum values that can be used for the externally connected capacitor?

_____

_____

24. What is the typical minimum pulse width the device can be wired to have?

_____

25. What is the equation for calculating the pulse width if external timing components are used and the capacitor's value is greater than 1000pF?

26. Calculate the pulse width if a 180kΩ external resistor and 68nF external capacitor is used.

Copyright: Not to be reproduced without permission

Student notes

Section 4 Counters

Purpose To develop your understanding of basic counter principles and to develop your ability to interpret the manufacturers' data sheets for common commercially available counter ICs.

Objectives At the end of this section you should be able to:

- Explain the basic operation of up and down counters (at the block diagram level only)
- Define the terms *overflow*, *weighting*, *least significant bit*, *most significant bit*, *modulus* and *decade counter* in terms of counter principles
- Calculate the modulus of a simple up/down counter given the number of bits in its output
- Calculate the highest number that a simple up/down counter can count to (both binary and decimal) given the number of bits in its output
- Draw and/or interpret the timing diagram of Mod-n up and down counters for any value of *n*
- Give typical applications for counter circuits
- Explain why counter circuits are also frequency dividers
- Interpret manufacturers' data for common commercially available counter ICs
- Explain the terms *presettability* and *cascadability* with regard to counter ICs
- Predict the modulus of IC counter circuits that use external combinational logic to have a modulus less than the IC has been designed for
- Use a logic probe and an oscilloscope to test and measure sequential logic ICs

Introduction

You have previously learnt about the operating principles of D-type and J-K flip flops. In addition, applications for these devices were listed including the counter. In this section we'll look at counter principles and the operation of a common commercially available counter IC.

Counter principles

In digital electronics a counter is a circuit or device that can count up and/or down in binary every time it is triggered.

The basic counter consists of a trigger input and two or more data outputs. The symbol for a basic three output (or 3-bit) counter is shown in Figure 1 below. (How counter circuits are actually implemented using flip flops is explained in Appendix 1.)

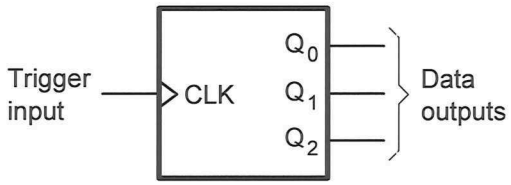


Figure 1 *The basic block diagram of a 3-bit counter*

As Figure 1 shows, the trigger input is usually called the clock (*CLK*). The clock is usually edge-sensitive and can be either positive edge-triggered or negative edge-triggered depending on the counter's design. A negative edge-triggered counter would have the circle on the clock input to denote this.

As Figure 1 also shows, the data outputs are often labelled Q_0, Q_1, Q_2, etc because these are the outputs of flip flops. Other labels used for counter outputs include Q_A, Q_B, Q_C and so on or just *A*, *B*, *C* and so on.

As each output represents a column of a binary number, the outputs are weighted. Output Q_0 has the lowest weighting (each "1" in this column is worth only one in decimal) and is said to be the *least significant bit* (LSB). On a 3-bit counter, output Q_2 has the highest weighting (each "1" in this column is worth four in decimal) and so it is said to be the *most significant bit* (MSB).

The simplest counter counts up by one each time it's clocked until it reaches the highest number in the count sequence (which is 111 for a simple 3-bit up-counter). On reaching the highest number, the next clock pulse causes the counter to output the lowest number (which is 000 for a 3-bit up-counter) and the counter counts up again on subsequent clock pulses. This is shown in Table 1 below. When an up-counter's output goes from the highest number to the lowest number on one clock pulse, the counter is said to *overflow*.

Table 1

Clock pulse	Count		
	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0
9	0	0	1
10	0	1	0

Counter *overflows*

Count sequence repeats

If you consider the count sequence in Table 1 above, you will see that the counter's output produces eight unique binary numbers. These are: 000, 001, 010, 011, 100, 101, 110 and 111. Once the counter overflows, this cycle is simply repeated. That being the case, there are eight counts in the count sequence and so the counter is said to have a *modulus* of eight. "Modulus" is a mathematical term simply meaning the number of different numbers used in an arithmetic system. As you have probably already discovered, we like abbreviations in electronics and so a counter with a modulus of eight is usually simply written and stated as being a *Mod-8* counter.

The modulus of simple counters is a function of the number of bits in the output. A 2-bit counter has a modulus of four, a 3-bit counter has a modulus of eight, a 4-bit counter has a modulus of sixteen and so on.

A simple counter's modulus can be calculated using the expression:

$$Mod = 2^n$$

Where n is the number of bits in the output

And, the highest number in the count sequence is:

$$Highest\ number = 2^n - 1$$

Practise using these equations by trying the following questions.

1. What is the modulus of the CMOS 4020 14-bit counter IC?

2. What is the highest number in binary and decimal that the 4020 can count to?

Care must be taken in deciding when to use these equations because, as you will see, there are ICs with *n* number of outputs that have a modulus less than 2^n. For example, there are several 4-bit counter ICs that only count from 0 to 9 and so have a modulus of ten and not sixteen (which 2^4 suggests they would have). These counters still need to have 4-bit outputs because the numbers 8 and 9 in binary are 4-bit numbers (ie 8 is 1000 and 9 is 1001).

Another simple counter is the one that counts down by one each time it is clocked until it reaches the lowest number (000 for a simple 3-bit counter). On reaching the lowest number, the next clock pulse causes the output to overflow (or sometimes referred to as *underflow* because it's counting down) and returns to the highest number ready to continue counting down. This is shown in Table 2 below.

Table 2

Clock pulse	Count		
	Q_2	Q_1	Q_0
0	1	1	1
1	1	1	0
2	1	0	1
3	1	0	0
4	0	1	1
5	0	1	0
6	0	0	1
7	0	0	0
8	1	1	1
9	1	1	0
10	1	0	1

Counter *overflows*

Count sequence repeats

The equation $Mod = 2^n$ can be used for calculating the modulus of simple down-counters and $2^n - 1$ gives the highest number that a down-counter counts down from.

Counter timing diagrams

The operation of a simple up and down counters can also be described using a timing diagram. Figure 2 below shows the timing diagram of a (negative edge-triggered) 3-bit up-counter whose outputs all start at logic-0.

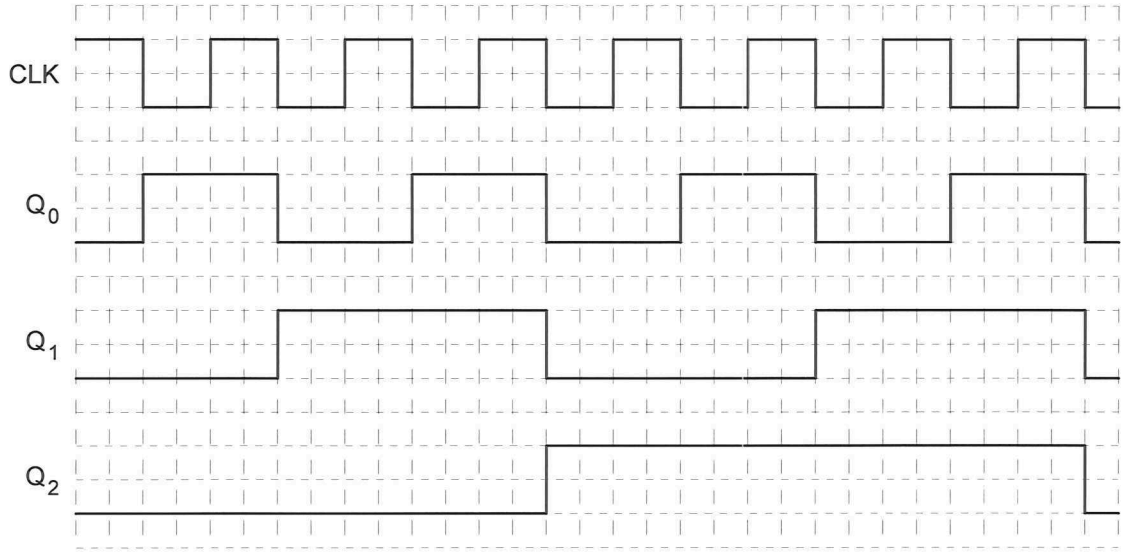


Figure 2

The fact that the signals on the outputs represent a count sequence might be a little clearer if the logic levels are written for each clock pulse as shown in Figure 3 below.

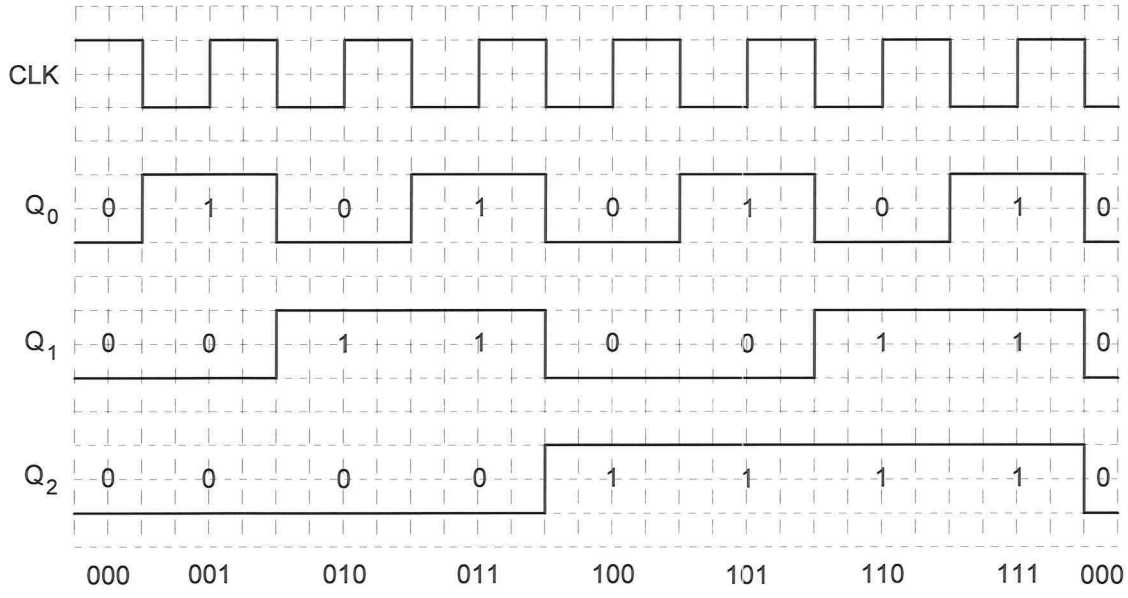


Figure 3

If you read-off each column of ones and zeros as a 3-bit number (remembering the Q_0 is the LSB and Q_2 is the MSB), you'll see that these toggling outputs are providing an up-count sequence.

Figure 4 below shows the timing diagram of a (positive edge-triggered) 3-bit down-counter whose outputs all start at logic-0 initially then all become logic-1 when the counter overflows on the first clock pulse.

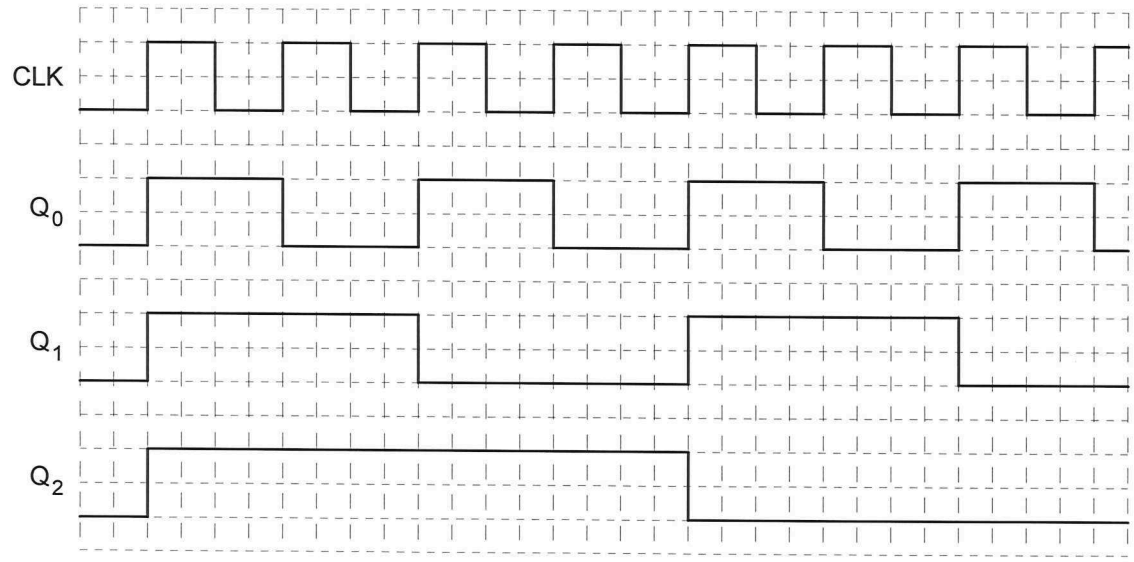


Figure 4

Notice that for both up and down counters the least significant bit changes logic level the most often and the most significant bit the least often.

Counter applications

The most obvious use for counter circuits is to count things. Examples include: the number of car spaces still available in a car park, the number of a person's turn at the delicatessen counter in a supermarket, the number of cars that have travelled along a road, and so on.

Automated process lines in factories also need counters. They count things like: the number of beers packed into each carton, the amount of soft-drink that is poured into each can, the number of biscuits in a Tim Tam tray and so on.

Perhaps a little less obvious, counters are also used in applications such as: digital tachometers (to count the number of revolutions of a motor per minute), digital odometers (to count the number kilometres travelled by a car), digital pedometers (to count the number of steps a person takes while walking), digital frequency counters (to count the number of cycles per second of an AC signal), and so on.

Counters are also crucial to the operation of digital clocks in equipment like clock radios, microwave ovens, PVRs and so on. In these examples one counter counts the seconds, another the minutes, still another counts the hours and a fourth counts the tens of hours.

Another important application for the counter that doesn't involve counting objects is the frequency divider (also known as a frequency scaler). Counters can be used to divide the frequency of a signal down by any whole number. This is true for both up and down counters and is clearly demonstrated in the timing diagrams in Figures 2 and 4 on the previous two pages.

If you compare the frequencies of the signal on Q_0 (the LSB) and the clock signal on both diagrams, you'll notice that Q_0 is half the clock's frequency. Similarly, the signal on Q_1 is half the frequency of the signal on Q_0 and the signal on Q_2 is half the frequency of the signal on Q_1. So the counter is definitely producing output signals that are divisions of the clock signal.

Importantly, these timing diagrams are for Mod-8 counters and if you compare the frequency of the signal on Q_2 (which is the MSB) with the clock you'll notice that Q_2 is an eighth of the clock's frequency. It is always the case that the MSB output of a Mod-n counter is one n^{th} of the clock frequency. For example, the MSB output of a Mod-5 counter is one fifth of the clock frequency (though the signal's duty cycle may be different to that of the clock's duty cycle).

Counter ICs

Across the range of commercially available TTL and CMOS digital ICs there are at least a dozen counter chips. These include the 7490, 7493, 4018, 74161, 74163, 74193, 4017, 4020, 4029 and the 4518. The availability of so many counter ICs reflects the importance and usefulness of counters to the operation of digital electronics systems.

You'll not be expected to recall the IC numbers for the theory tests. However, you are expected to know about counter IC characteristics including: *presettability* and *cascadability*. You're also expected to be able to predict the modulus of IC counter circuits that use external combinational logic to have a modulus less than the IC has been designed for. To explain the basics of these counter IC attributes, we'll use just one counter chip and discuss its operation in detail.

The 74193 4-bit, binary up/down counter

The 74193 is a 4-bit, positive edge-triggered, up/down-counter that is presettable and has an asynchronous clear input. (Note: The manufacturer also describes this device as being a synchronous counter. Refer to Appendix 1 for an explanation of the difference between asynchronous and synchronous counters. You're not expected to explain this difference in the exams.) The pinout for the 74193 is shown in Figure 5 below.

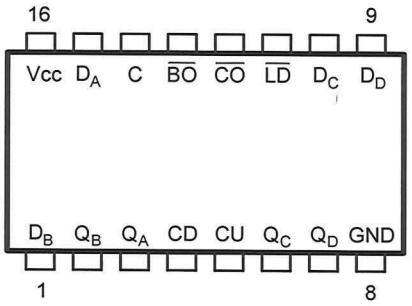


Figure 5 *Pinout data for the 74193*

To help make the circuit drawings using the 74193 a little easier to interpret, it has been redrawn in Figure 6 below in block diagram form with the pins arranged more conveniently.

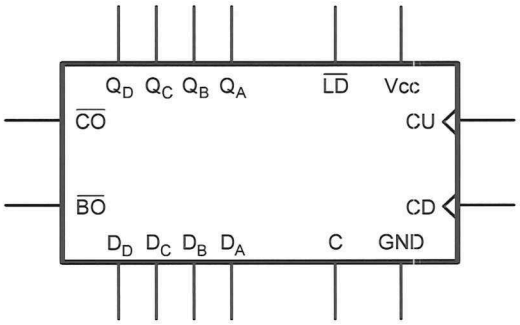


Figure 6 *A block diagram for the 74193*

If you look at Figure 6 you'll notice that the device has four outputs (Q_A to Q_D) and so it is a 4-bit counter. This means that it is capable of counting up from 0000 to 1111 (or down from 1111 to 0000) and so it has a modulus of sixteen. However, the manufacturers prefer to call their IC a 4-bit counter rather than a Mod-16 counter. This probably because it can be easily modified for a modulus less than sixteen and can be easily combined with other 74193 counters for a modulus greater than sixteen (as you will see).

To control whether the device counts up or down, it has two clock inputs (both of which are positive edge-triggered), one devoted to counting up (*CU*) and one devoted to counting down (*CD*). Obviously a counter cannot count up by one and down by one at exactly the same time so the two clocks are not allowed to be triggered simultaneously. The unused clock input must be made logic-1 to prevent it from interrupting the counter's expected operation (an effect often called *false triggering*).

As you might expect, the device has a *clear* input (*C*) which is active high and asynchronous. Applying a logic-1 to this input immediately resets all the Q-outputs (notwithstanding propagation delay). The clear input also over-rides the clock inputs. That is, if the device is clocked (either up or down) while the clear input is asserted, the clock is ignored. If the clear input is not used by the circuit then it should be hard-wired to ground (logic-0) to prevent false clearing.

The device's outputs are presettable which means that they can be forced to output any 4-bit binary number at any time. This is done using the data inputs (D_A to D_D) and is synchronised by the load ($\overline{LD}$) input. When the load input is asserted (by making it logic-0) the binary number on the data inputs is immediately transferred to the Q-outputs (notwithstanding propagation delay). D_A goes to Q_A, D_B goes to Q_B and so on. Like the clear input, the load input over-rides the clocks so, if the device is clocked (either up or down) while the load input is asserted, the clock is ignored. If the counter's preset option is not used by the circuit then the load input should be hard-wired to supply (logic-1) to prevent false presetting.

Figure 7 shows the minimum wiring requirements to operate the 74193 as a simple Mod-16 up-counter. Notice that the *CD* and $\overline{LD}$ pins are hard-wired to 5V to tie them high and the *C* pin is hard-wired to ground to tie it low.

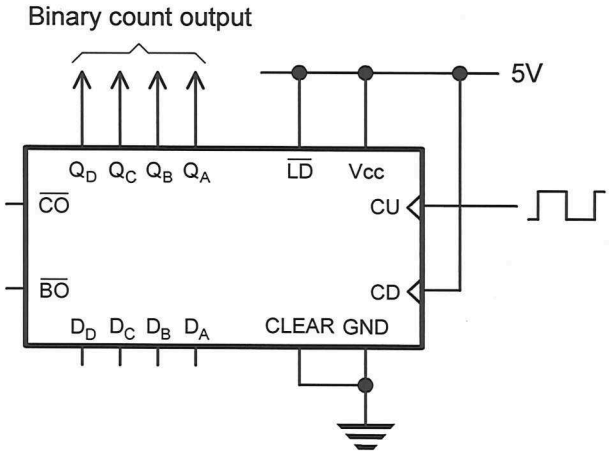


Figure 7 *The minimum wiring requirements to operate the 74193 as a simple Mod-16 up-counter*

Figure 8 shows the minimum wiring requirements to operate the 74193 as a simple Mod-16 down-counter. Notice that this time the *CU* pin is hard-wired to 5V to tie it high and the clock signal is applied to the *CD* pin instead. (To make the diagram a little simpler, the *CLEAR* input has simply been written as *C*.)

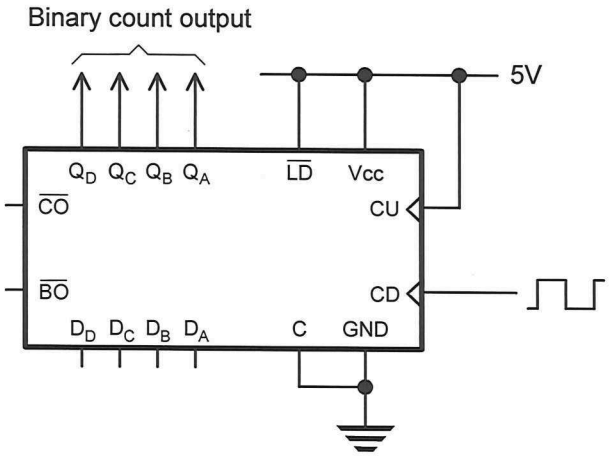


Figure 8 *The minimum wiring requirements to operate the 74193 as a simple Mod-16 down-counter*

Increasing the modulus of counter ICs

Counter ICs can be readily connected together (or cascaded) to make counters with more output bits. Having more output bits means the counter can have a bigger modulus and so can count to bigger numbers. For example, a 4-bit (Mod-16) counter can only count up to 1111 (or 15 in decimal) but cascading two of them produces an 8-bit (Mod-256) counter that can count up to 11111111 (or 255 in decimal). Similarly, three 4-bit counter ICs can be cascaded to make a 12-bit counter and so on.

External combinational logic circuitry is often needed to cascade counter ICs but the designers of the 74193 have made it so that external circuitry is unnecessary by building in to the device two extra outputs labelled $\overline{CO}$ for "carry out" and $\overline{BO}$ for "borrow out".

These outputs are usually logic-1. When the 74193 is made to count up, the $\overline{CO}$ pin goes to logic-0 when the counter's output changes to 1111. Then, on the next clock pulse when the counter overflows (that is, goes from 1111 to 0000) the $\overline{CO}$ pin returns to logic-1. The timing of this is shown in Figure 9 below.

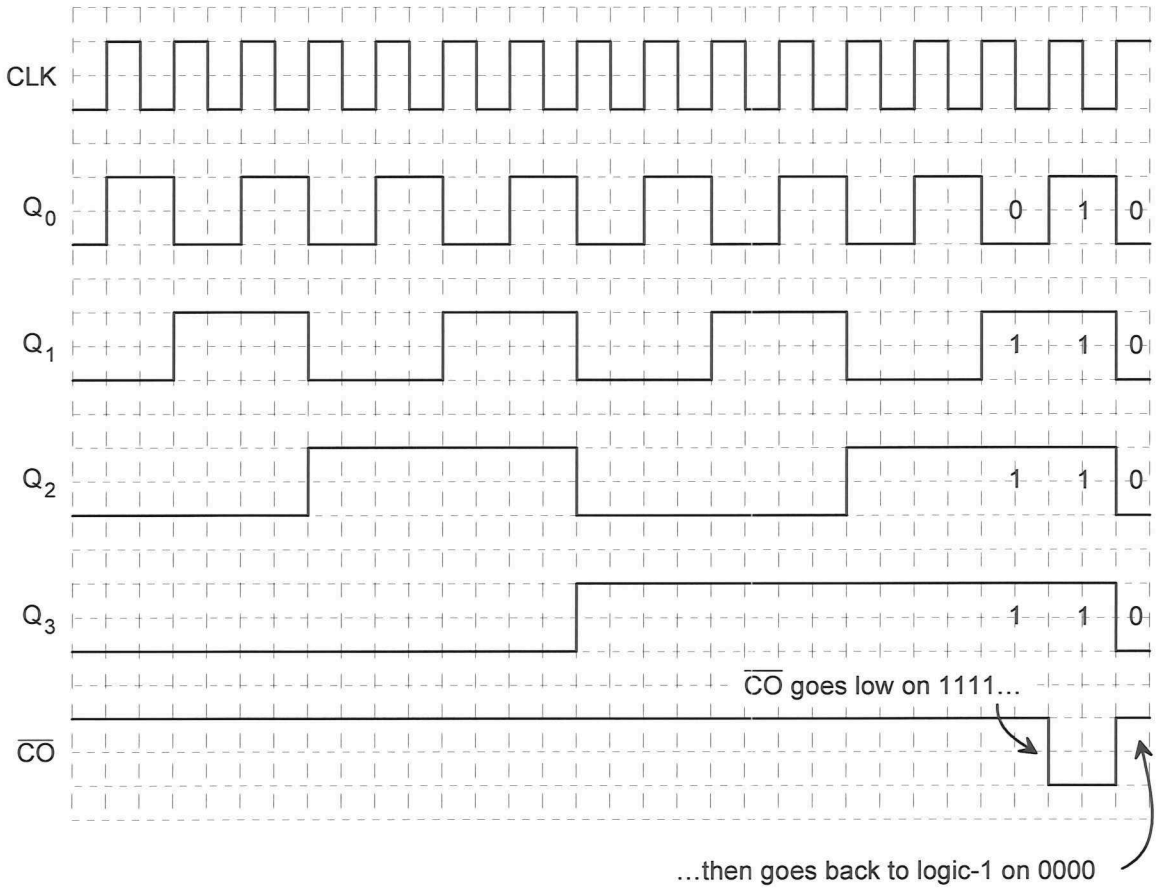


Figure 9

The $\overline{CO}$ output's action is used to make a simple 8-bit (Mod-256) up-counter using two 74193 ICs as shown in Figure 10 below.

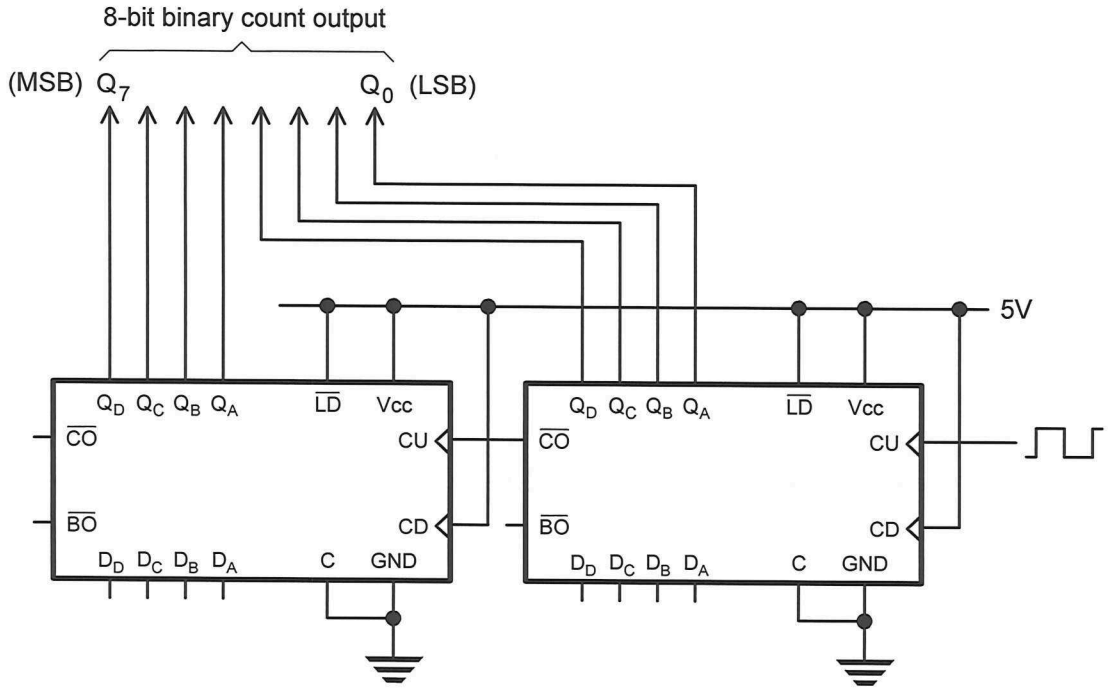


Figure 10 *Two 74193 counters cascaded to make a simple Mod-256 up-counter*

In this arrangement, the right side counter makes up the lower four bits of the 8-bit output number (Q_0 to Q_3) and the left side counter makes up the upper four bits of the 8-bit output number (Q_4 to Q_7).

Importantly, notice that the $\overline{CO}$ of the right side counter connects to the CU input of the left side counter. This means that every time the right side counter overflows, the $\overline{CO}$ output changes from logic-0 back to logic-1 triggering the left side counter causing it to count up one. This ensures that the 8-bit number that follows 00001111 is 00010000 and the 8-bit number that follows 00011111 is 00100000 and so on.

The $\overline{BO}$ output is used for exactly the same purpose as the $\overline{CO}$ output but for cascaded down-counters. The circuit of a simple 8-bit (Mod-256) down-counter is shown in Figure 11 below.

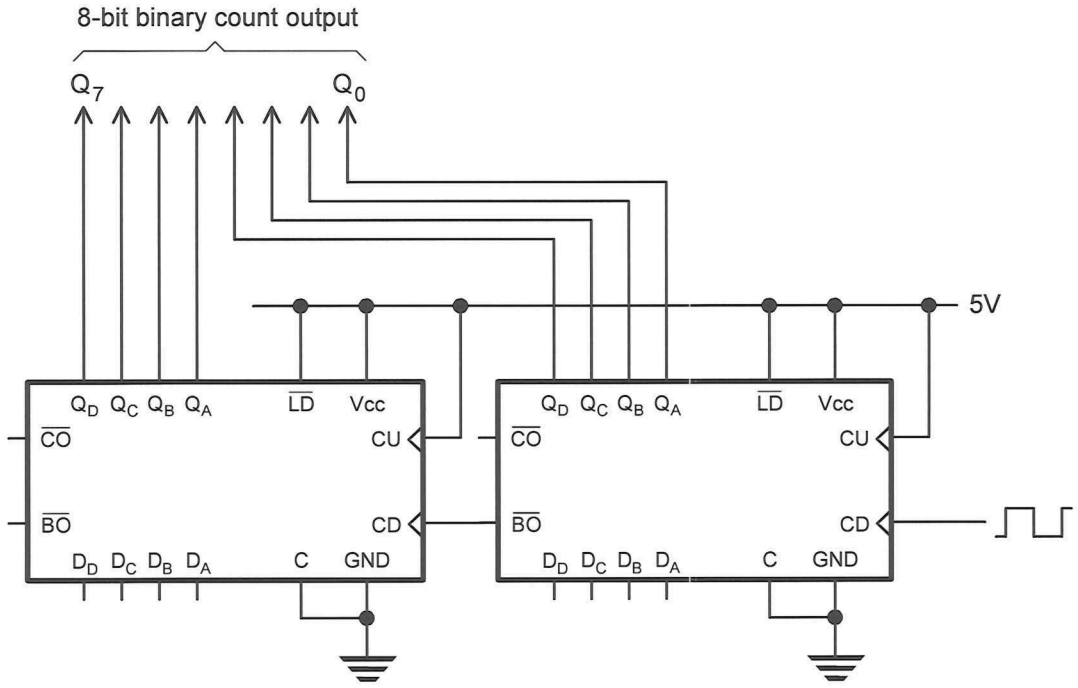


Figure 11 *Two 74193 counters cascaded to make a simple Mod-256 down-counter*

Decreasing the modulus of counter ICs

Many commercially available ICs are either Mod-10 (often called *decade* counters) or Mod-16. These are moduli (pronounced modu-lee) that are used in the operation of many digital systems so it is commercially viable to develop and manufacture large quantities of these counter chips.

Counters with other moduli are not as widely used and so there is less commercial benefit for manufacturers to provide such ICs. This is not a problem because it's possible to modify counters to implement a modulus less than the modulus they're manufactured to have.

A common method of doing this that can be used on any counter with a clear input is to decode the outputs (often requiring some external combination logic) and using the signal to reset the counter when it reaches the number that is one higher than the desired modulus.

This is probably better understood with some examples. Figure 12 below shows how the 74193 counter IC can be modified from the simple Mod-16 up-counter in Figure 7 (on page 4-11) to a Mod-8 up-counter.

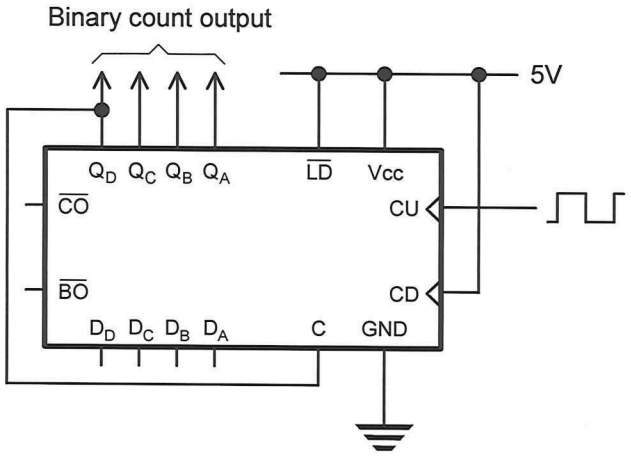


Figure 12 *The circuit of a Mod-8 up-counter using the 74193*

Notice that Q_D is connected to the IC's clear input. Recall that the clear input on a 74193 is active high and asynchronous so the moment that Q_D becomes a logic-1 the counter's outputs are all reset. That being the case, the counter counts up from 0000 to 0111 as normal. Then on the next clock pulse the output changes to 1000 making Q_D a logic-1 for the first time. This forces the counter to reset its outputs so the count immediately returns to 0000. Importantly, the number 1000 exists only for the time it takes for the device to reset (which is the clear input's propagation delay).

From this description we can see that there are nine numbers in the count sequence (0000 to 1000) but, as the ninth number exists for such a short time (you certainly wouldn't see the number on a display), it can be ignored for most applications. So in this way the circuit implements a Mod-8 counter. The circuit is unambiguously a Mod-8 counter if you consider that there are only eight clock pulses in each count cycle (the numbers 1000 and 0000 share the period of one clock pulse though 1000 would exist for a small percentage of it).

Figure 13 below shows the 74193 modified from a simple Mod-16 up-counter to a Mod-9 up-counter. Notice that this time some external circuitry is used to decode the counter's outputs and reset the device.

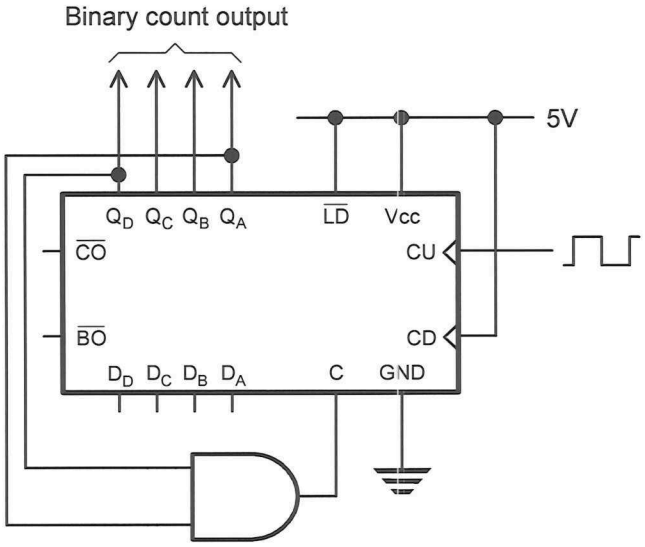


Figure 13 *The circuit of a Mod-9 up-counter using the 74193*

Recall that the clear input to the counter is active high. This means that to reset the counter to 0000 the AND gate's output must go to logic-1. As you know, to make the output of an AND gate logic-1 both of its inputs must also be logic-1 at the same time. Given that the AND gate's inputs are connected to Q_A and Q_D, this will happen for the first (and only) time on the number 1001. In other words, the counter counts from 0000 to 0001, 0010, 0011, 0100, 0101, 0110, 0111 and 1000 (which is nine numbers) then, when it attempts to output the number 1001 it is quickly reset back to 0000 to start the count sequence again.

Practise identifying the modulus of counters with feedback by trying the following questions.

1. What is the modulus of the up-counter in Figure 14 below?

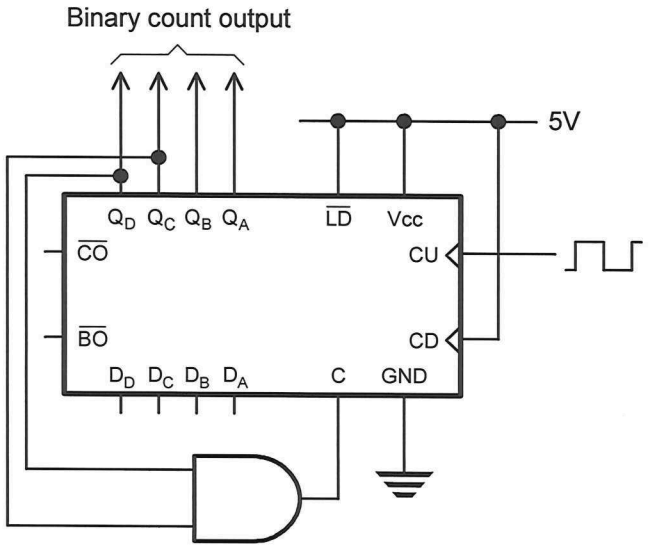


Figure 14

2. What is the modulus of the up-counter in Figure 15 below?

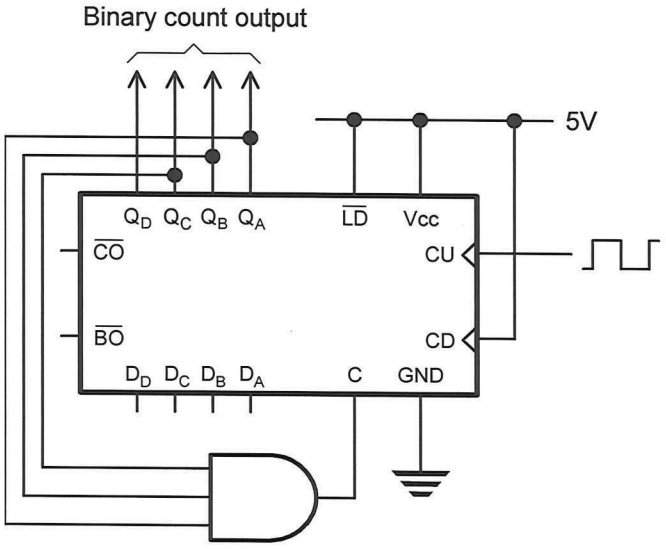


Figure 15

As the circuits in Figures 13, 14 and 15 show, when lowering the 74193 counter's modulus, sometimes it is necessary to use external circuitry. There are two problems with this. First, extra circuitry means extra gates and therefore extra ICs. This increases component costs and circuit board size.

It's very easy to implement a counter using a microcontroller (provided you have the coding software and an IC programmer). Using a microcontroller for this purpose permits counters with any modulus up to the maximum for the IC's number of output bits. Moreover, the count sequence wouldn't include the temporary number (ie the number that is one higher than the highest number in the count sequence) that's needed to reset the counter. This would be advantage where this spurious count would upset the operation of the digital circuits connected to the counter's output.

Skill practice 4

Pratcise testing and measuring sequential logic digital counter ICs

This exercise is practise for the sorts of skills you may be required to perform in a practical test. Remember, in any practical tests you will be working alone so make sure that you can perform all the steps. It should take you about 1½ hours to complete this exercise.

Equipment

- Digital trainer
- logic probe
- two 74LS193
- 74LS08
- BNC to alligator-clip lead
- Attenuated CRO probe lead
- hook-up wire (terminated)
- IC extractor

Remember:

Follow TAFE NSW WHS guidance at all times!

 Copyright: Not to be reproduced without permission

Work tasks

1. Read your WHS responsibilities at the top of the form below. Then conduct a WHS risk assessment and record your findings in the space provided.

Responsibilities of students under the Model WHS Act: s28

- Take reasonable care for your own health and safety by working safely at all times
- Take reasonable care to ensure that your acts or omissions don't put the health and safety of others at risk
- Follow all TAFE NSW WHS guidance and comply with all reasonable instructions from TAFE NSW staff to assist them in complying with the TAFE NSW WHS requirements
- In addition to the above, you must:
  - use and maintain machinery, tools and all other equipment properly and safely
  - ensure that your work area is free of hazards
  - notify a TAFE NSW staff member of actual or potential hazards
  - wear/use prescribed safety equipment
  - take notice of any safety signs and adhere to their instructions

Risks involved in this activity include:	**Control measures:**
■ Trip hazards (eg students bags) ■ Objects dropped on feet (while equipment is taken to and from benches).	■ Move bags and other objects from walkways ■ Plan lifting of equipment
Others: _____	Other: _____
_____	_____
_____	_____

My signature here indicates that I have read and understand my responsibilities under the Model WHS Act s28 (detailed above). I have also conducted a risk assessment before undertaking this activity and have identified measures to control these risks and have implemented them.

Signature: _____ Date: _____

2. Gather the equipment needed for this exercise.

3. Read the following points and make sure that you perform them when wiring all circuits:

   - Write the IC pin numbers of the devices you work with on the circuit that you're instructed to wire.
   - Handle all ICs carefully - remember ESD can damage them.
   - Ensure the power supply is connected to all ICs in the circuit.
   - When plugging hook-up wire into the breadboard to make connections to the IC's pins, use holes that are furthest away from the pin.

4. Wire the circuit of Figure 1 on the digital trainer. Pinout data for the 74LS193 can be found in Appendix 6.

   Note: Use the *B* output of the P_B pulser button for the flip flop's clock input

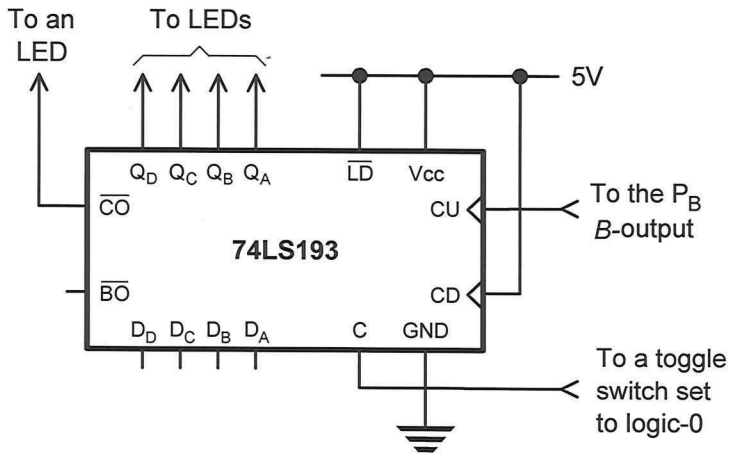


Figure 1

5. Turn on the trainer.

6. Trigger the counter several times using the pulser and observe the outputs.

   Self check: Each time you press the pulser the LED display should increment by one in a count sequence. If it doesn't, double-check your wiring. If it still doesn't, call the teacher.

7. Throw the clear input's toggle switch up and down once and observe the effect on the outputs.

Question 1
From your observation, which of the following statements best describes the operation of the 74LS193's clear input?

- ☐ Active low and asynchronous
- ☐ Active high and asynchronous
- ☐ Active low and synchronous
- ☐ Active high and synchronous

8. Press the pulser once and observe the Q and $\overline{CO}$ outputs. Record their logic levels in Table 1 below. Then repeat this until the table is complete.

Table 1

No. of clock pulses	Count outputs				$\overline{CO}$ output
	Q_D	Q_C	Q_B	Q_A	
0	0	0	0	0	1
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					

Question 2
What type of counter is the circuit?

☐ An up-counter

☐ A down-counter

Question 3
How many numbers are there in the count sequence before it repeats?

_____

Question 4
What is the counter's modulus?

_____

Question 5
When does the $\overline{CO}$ output change from a logic-0 to a logic-1?

_____

Question 6
What can the $\overline{CO}$ output be used for?

_____

_____

The teacher needs to check your work at this point...

9. Turn off the trainer and wire the circuit of Figure 2.

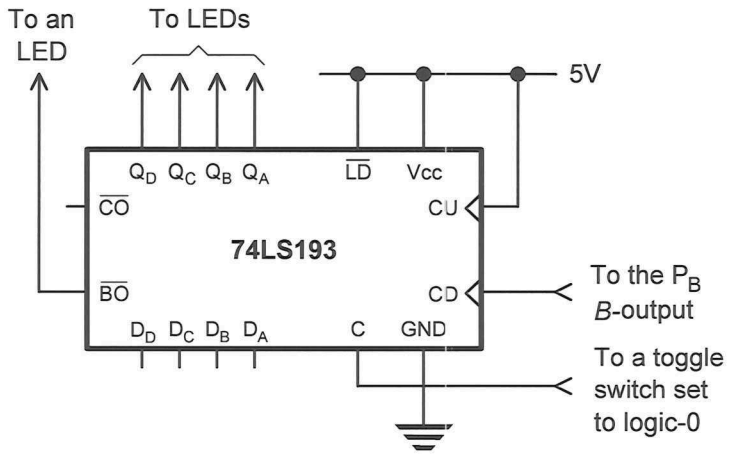


Figure 2

10. Turn on the trainer and clear the counter.

11. Use the pulser to clock the counter through a complete count sequence.

12. Repeat step 11 but this time note the number of counts in the count sequence and the operation of the $\overline{BO}$ output.

Question 7
What type of counter is the circuit?

☐ An up-counter

☐ A down-counter

Question 8
What is the counter's modulus?

_____

Question 9
When does the $\overline{BO}$ output change from a logic-0 to a logic-1?

_____

Question 10
What can the $\overline{BO}$ output be used for?

The teacher needs to check your work at this point...

13. Turn off the trainer and wire the circuit of Figure 3. Pinout data for the 74LS08 can be found in Appendix 6.

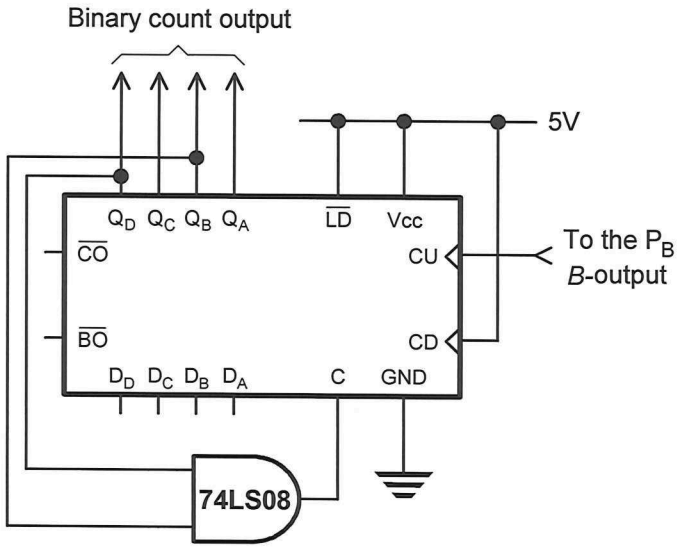


Figure 3

14. Turn on the trainer.

15. Use the pulser to clock the counter until it overflows.

16. Continue to clock the counter through a complete count sequence noting the number of counts in the count sequence.

 Copyright: Not to be reproduced without permission

Question 11
What type of counter is the circuit?

☐ An up-counter

☐ A down-counter

Question 12
What is the counter's modulus?

_____

Question 13
What is another name for counters with this modulus?

_____

Question 14
Briefly explain how the 74LS193's modulus has been changed from sixteen to ten.

_____

_____

_____

The teacher needs to check your work at this point...

17. Disconnect the *CU* input from the pulser and connect it to the trainer's in-built function generator's ***TTL Mode*** output.

18. Set the trainer's *Range* control to the *×1* position.

19. Turn the trainer's *Frequency* control to the "10" setting.

    Self check: The counter should now be repeatedly counting from 0000 to 1001 on its own. If not, check that the counter's *CU* input is correctly connected to the trainer's *Clock* output. If the counter is still not counting on its own, call the teacher.

20. Turn the trainer's *Range* control to the *×1k* position.

21. Set up the bench-mounted CRO to display both channels. Adjust the *Horizontal Timebase* control to the *200µs/div* setting and the Channel 1 *Vertical Attenuation* control to the *2V/div* setting.

22. Use the BNC to alligator-clip lead to connect the CRO's Channel 1 input to the *CU* input on the counter.

23. Check that you obtain a signal on the CRO's display consisting of twenty or more cycles across the screen.

    Note: Don't adjust the CRO's timebase to spread the cycles out - You'll see why shortly.

24. Set the *Attenuated CRO Probe* to the *×10* position.

    Note: This probe will be connected to one of the counter's outputs and it has been set to the *×10* position to avoid loading. However, doing so cuts the signal's amplitude to one tenth of its actual size.

25. Set the CRO's Channel 2 *Vertical Attenuation* control to the *0.2V/div* setting.

26. Use the probe to connect the CRO's Channel 2 input to the Q_D output of the counter.

    Self check: You should now see two pulse trains on the CRO's screen. If you don't call the teacher for assistance.

27. Use the CRO to determine the frequency of the clock signal. Record your measurement in Table 2 on the next page.

28. Use the CRO to determine and record the frequency of the signal on the counter's Q_D output.

Copyright: Not to be reproduced without permission

Table 2

Frequency of the *Clock* signal	Frequency of the Q_D signal

Question 15
How many cycles are there in the clock signal for every one cycle of the signal on Q_D?

Question 16
Write the ratio of Q_D cycles to clock cycles as a fraction (eg $\frac{Q_D \text{ cycles}}{\text{Clock cycles}}$).

Question 17
What's the significance of the ratio that you obtained in your answer to the previous question in relation to the counter's modulus?

The teacher needs to check your work at this point...

Review questions

Answer these questions to check your understanding of what you have learnt from this chapter. Doing this will also help to prepare you for the assessments.

1. In terms of digital electronics, what is a *counter*?

2. What happens when an up-counter *overflows*?

3. What happens when a down-counter *overflows*?

4. What do the letters LSB stand for with regard to counters?

5. What do the letters MSB stand for with regard to counters?

6. What does it mean when the MSB is said to be the highest *weighted* output?

7. What does the term *modulus* describe with regard to counters?

 Copyright: Not to be reproduced without permission

Questions 8 to 16 refer to the simple block diagram of the counter in Figure 1

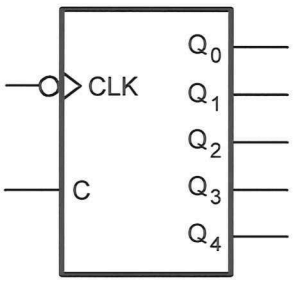


Figure 1

8. How many bits make up the counter's output?

- ☐ 7
- ☐ 5
- ☐ 4
- ☐ None of the above

9. The clock input is

- ☐ positive edge-triggered.
- ☐ negative edge-triggered.
- ☐ active high enabled.
- ☐ active low enabled.

10. The clear input is

- ☐ positive edge-triggered.
- ☐ negative edge-triggered.
- ☐ active high enabled.
- ☐ active low enabled.

11. The clear input is likely to be

    ☐ asynchronous.
    ☐ synchronous.

12. Which output is the LSB?

    _____

13. Which output is the MSB?

    _____

14. What is each logic-1 on the MSB worth in decimal?

    _____

15. What is the counter's modulus (assuming it is a simple up-counter)?

    _____

    _____

16. What is the highest number (in both binary and decimal) that the counter can count up to?

    _____

    _____

    _____

 Copyright: Not to be reproduced without permission

Questions 17 and 18 refer to the timing diagram of a counter in Figure 2 below

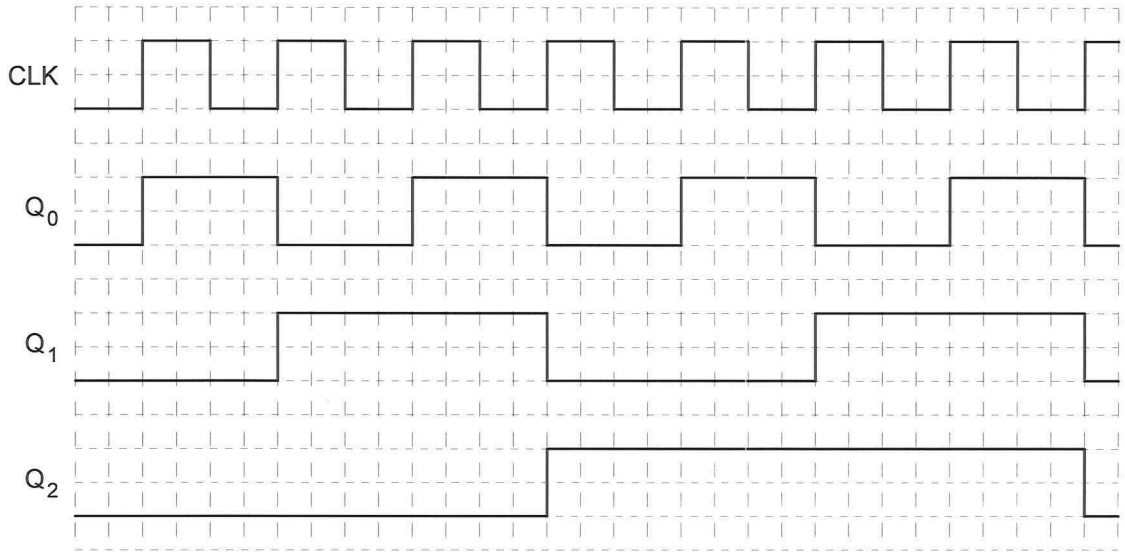


Figure 2

17. Is the counter positive edge-triggered or negative edge-triggered? Explain your answer.

18. Is the counter an up-counter or a down-counter? Explain your answer.

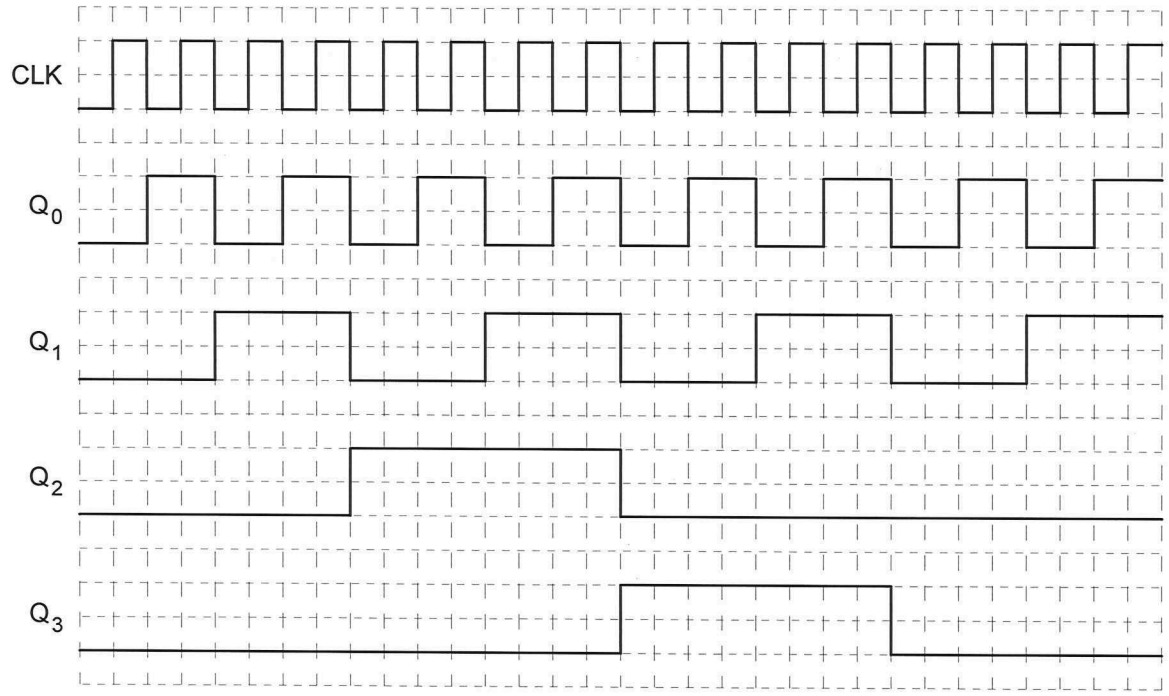


Figure 3

19. What is the modulus of a counter that has the timing diagram shown in Figure 3. Explain your answer.

Copyright: Not to be reproduced without permission

20. Calculate the likely modulus of a counter with ten output bits.

21. Why is this only the "likely" modulus of the IC?

22. What is a cascaded counter?

23. What is the purpose of cascading counter ICs?

Questions 24 to 28 refer to the 4017 IC. Data for the device can be found in Appendix 6

24. How many bits make up the counter's output?

- ☐ 9
- ☐ 10
- ☐ 11

25. The IC can count

- ☐ up only.
- ☐ down only.
- ☐ both up and down.

26. The clock input is

- ☐ positive edge-triggered.
- ☐ negative edge-triggered.
- ☐ active high enabled.
- ☐ active low enabled.

27. The reset input is

- ☐ positive edge-triggered.
- ☐ negative edge-triggered.
- ☐ active high enabled.
- ☐ active low enabled.

28. What is the purpose of the C_{out} pin?

 Copyright: Not to be reproduced without permission

Questions 29 to 32 refer to the 4029 IC. Data for the device can be found in Appendix 6

29. How many bits make up the counter's output?

- ☐ 4
- ☐ 8
- ☐ 10

30. The IC can count

- ☐ up only.
- ☐ down only.
- ☐ both up and down.

31. The clock input is

- ☐ positive edge-triggered.
- ☐ negative edge-triggered.
- ☐ active high enabled.
- ☐ active low enabled.

32. What is the purpose of the inputs labelled P_0 to P_3?

Section 5 Shift registers

Purpose To develop your understanding of basic shift register principles and to develop your ability to interpret the manufacturers' data sheets for common commercially available shift register ICs.

Objectives At the end of this section you should be able to:

- Explain the difference between serial and parallel data transfer
- State the comparative advantages and disadvantages of the two data transfer systems
- Explain the basic principle of operation of the four shift registers (at the block diagram level only)
- List applications for the four shift registers
- State the alternate name for a parallel in, parallel out shift register
- Interpret manufacturers' data for common commercially available shift register ICs
- Use a logic probe to test and measure sequential logic ICs

Introduction

You may recall that flip flops form the basis of many sequential logic circuits and Appendix 1 shows how they are used to make counters. Flip flops are also crucial to the operation of another set of sequential logic circuits called *shift registers*. In this section we'll look at the different types of shift register and examine the operation of a useful shift register IC.

Serial and parallel data transfer

Before discussing shift registers we must discuss the different ways in which digital data like binary numbers and codes can be transferred from one point to another both within digital systems (for example, inside a computer) and between digital systems (for example, between two computers).

Data can be transferred in either one of two forms: serial or parallel. When data is transferred in serial form, the bits that make up the number or code are transferred one at a time, one after the other along a single wire. Figure 1 shows the 4-bit binary word 1001 being sent this way.

Figure 1 *A 4-bit data word being transmitted in serial form along a single wire*

When data is transferred in parallel form, the bits that make up the number or code are transferred all at once along the same number of wires as there are bits. Figure 2 shows the same 4-bit binary word being sent this way.

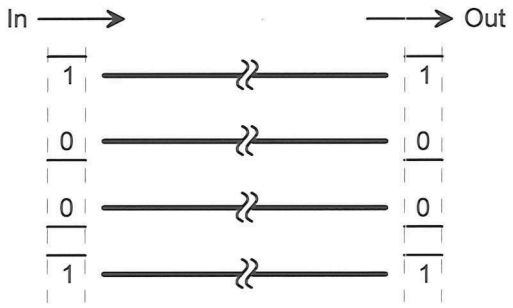


Figure 2 *A 4-bit data word being transmitted in parallel form along four wires*

In terms of timing, parallel data transfer is generally faster than serial. That's because it takes only one clock pulse to send parallel data regardless of the number of bits. Serial data transfer requires as many clock pulses as there are bits (for example, four clock pulses are needed to send a 4-bit number).

Parallel data transfer is usually used to move data between system components within a digital system like when transferring data from the microprocessor to RAM inside a computer. However, parallel data transfers requires more wires than serial data transfer. So, if the data transfer is to occur over any distance, serial transfer is likely to be cheaper and more convenient so it is usually preferred. Serial data transfer is used in several applications including USB (Universal Serial Bus) connections between computers and peripheral devices like printers, scanners and laptop computers.

Shift registers

Any register is just a store of information. A good example is the class rollbook which is a register that contains information about students (ie their names and attendance). Registers in digital electronics then are a store of digital information, either binary numbers or codes.

Importantly, as data can be transferred in either serial or parallel form, registers can be designed to handle both. So, one type of register inputs and outputs data in serial form and another inputs and outputs data in parallel form. In addition to this, there are two other registers that work with both serial and parallel data. One inputs data in serial form and outputs it in parallel and the other inputs data in parallel and outputs it in serial. This gives us a total of four registers.

In the normal operation of three out of four of these registers, the bits that make up the data get moved or "shifted" within the circuit from one flip flop to the next in a digital electronics version of the children's game *pass-the-parcel* and so the four registers are collectively known as shift registers.

This shifting action can be demonstrated if we examine the operation of the simple 3-bit shift register in Figure 3 below.

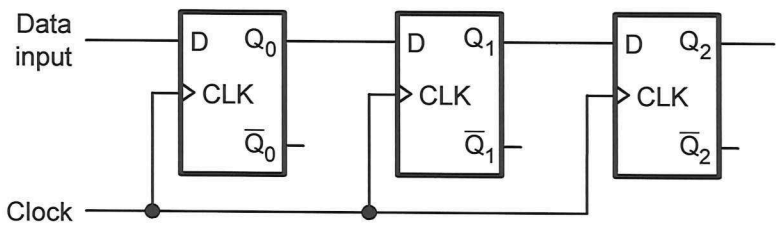


Figure 3 *A simple 3-bit shift register*

You'll notice that the circuit is made up of cascaded D-type flip flops with all of the clock inputs connected together. (Note: The circuit shouldn't be confused with a counter circuit which is made out of cascaded toggling flip flops.) The D-type flip flops are positive edge-triggered in this example but the circuit will work with negative edge-triggered devices. Also, J-K flip flops can be used instead of D-type flip flops but they have to be wired to operate as D-type flip flops.

To explain the circuit's operation, assume it has just been turned on. The outputs of the flip flops almost immediately become either logic-0 or a logic-1 but, as they have not been determined by the action of the circuit's operation, they're meaningless. This is shown below in Figure 4 with the meaningless flip flop output values represented by the letter X.

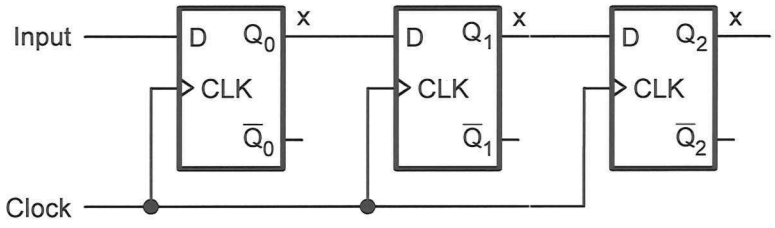


Figure 4

Now, let's suppose a logic-1 is applied to the circuit's input and the circuit is clocked for the first time. The clock causes the logic levels on the flip flips' D-input to be copied to their output. However, as the first flip flop is the only one with meaningful data on its input (the input to the other flip flops are still just the initial random values on the output of the flip flop they're connected to), only the first flip flop's output is meaningful. The new circuit conditions are shown in Figure 5 below.

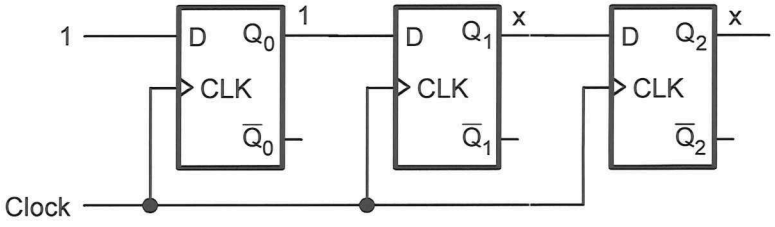


Figure 5

For the purposes of contrast, let's assume that the data input is now made logic-0. On the next clock pulse this logic-0 is transferred to the first flip flop's output. But, due to propagation delays, the input to the second flip flop at the time of the clock pulse is still a logic-1 so its output becomes a logic-1. The new circuit conditions are shown in Figure 6 below.

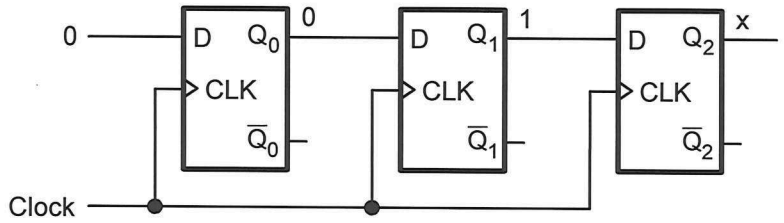


Figure 6

If the data input is kept a logic-0 when the third clock pulse triggers the flip flops the conditions around the circuit change as shown in Figure 7 below.

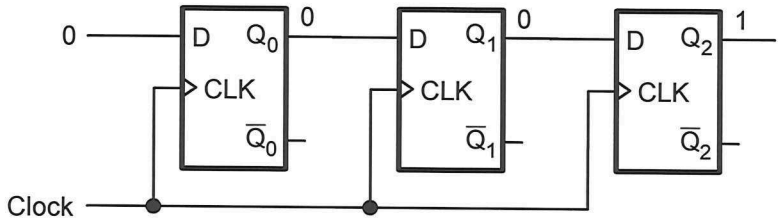


Figure 7

From this example we can see that data (that is, logic-1s and logic-0s) shift from one D-type flip flop to the next one at a time with each successive clock pulse.

Shift register ICs

Across the range of commercially available TTL and CMOS digital ICs there are probably almost as many shift register chips as there are counter chips. This is because shift registers are as useful and important to the operation of digital electronics systems as counters. Shift register ICs include the 74164, 74165, 74194, 4006, 4014, 4015 and the 4021.

You'll not be expected to recall the IC numbers for the theory tests. However, you are expected to know the basic principles of operation of the four basic types of shift register. These are: serial in, serial out; serial in, parallel out; parallel in, parallel out; and parallel in, serial out. To explain the basics of these shift registers, we'll use just one shift register IC and discuss its operation in a little detail.

The 74194 4-bit bi-directional universal shift register

This TTL device is a positive edge-triggered 4-bit shift register that can shift data both left and right. Importantly, being a *universal* shift register, the 74194 can implement the four shift register combinations named above. The pinout for the 74194 is shown in Figure 8 below.

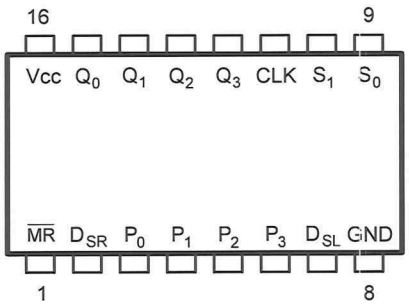


Figure 8 *Pinout data for the 74194*

To help make the device's operation a little easier to understand, it has been redrawn in Figure 9 below in block diagram form with the pins arranged more conveniently for drawing circuits.

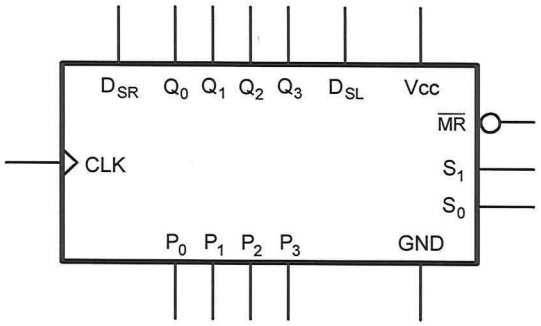


Figure 9 *A block diagram for the 74194*

Drawn this way we can see that the 74194 has four outputs (Q_0 to Q_3) and so it is a 4-bit shift register. An important point to make here is that the outputs of shift registers are not mathematically weighted like the outputs of counters. This is because shift registers work with codes as well as numbers.

The 74194 has one clock input which is positive edge-triggered. The IC also has a clear input which is asynchronous and active low. On this device the clear input is called the *master reset* because when it is asserted it over-rides all other input pins and ignores them. If the master reset is not being used then it should be tied high to prevent false resetting of the device.

To control the 74194's mode of operation, two *mode select* inputs are provided and are labelled S_0 and S_1. The relationship between the logic levels on the select inputs and the device's mode of operation is shown in Table 1 below.

Table 1

Select inputs		Operating mode	Outputs			
S_1	S_0		Q_0	Q_1	Q_2	Q_3
0	0	Hold	Q_0	Q_1	Q_2	Q_3
0	1	Shift right	D_{SR}	Q_0	Q_1	Q_2
1	0	Shift left	Q_1	Q_2	Q_3	D_{SL}
1	1	Parallel load	P_0	P_1	P_2	P_3

The D_{SL} and D_{SR} inputs are used for serial data inputs and two are needed because the device is capable of shifting data right and left. This will become more obvious when we look at some circuits.

The 74194 operated as a serial in, serial out shift register

Figure 10 below shows the minimum wiring requirements to operate the 74194 as a 4-bit serial in, serial out shift (right) register.

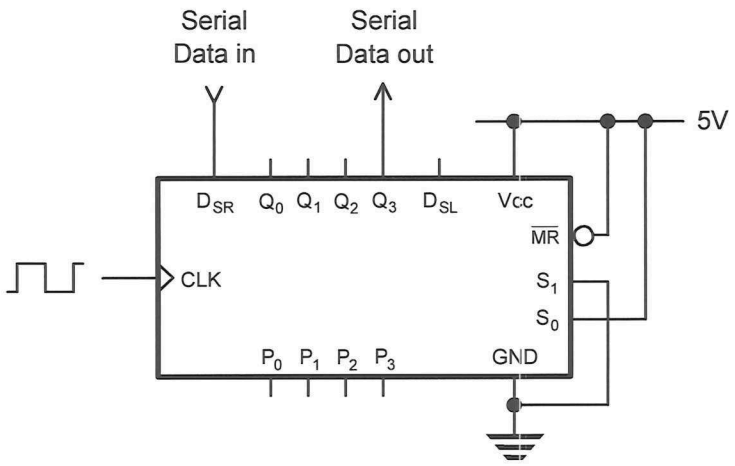


Figure 10 *The minimum wiring requirements to operate the 74194 as a 4-bit serial in, serial out shift (right) register*

Every time the clock triggers the device, the logic level on the D_{SR} input is read and transferred to the Q_0 output. At the same time, the logic levels on the other Q-outputs move along to the next Q-output on their right. Data is available on the Q_3 output in serial form four clock pulses after it was entered into the shift register in serial form via the D_{SR} input.

To demonstrate the operation of the 74194 as a serial in, serial out shift register, the flow of data through the circuit over 5 clock pulses starting with a logic-1 for the input data followed by logic-0, 0, 1 and 0.

Initially, a logic-1 (denoted using an asterisk to show that it's the first bit) is applied to the data input. As the device has not yet been clocked, the logic levels on the outputs have not been determined by the circuit and are meaningless. For this reason they're represented by the letter X (see Figure 11 below).

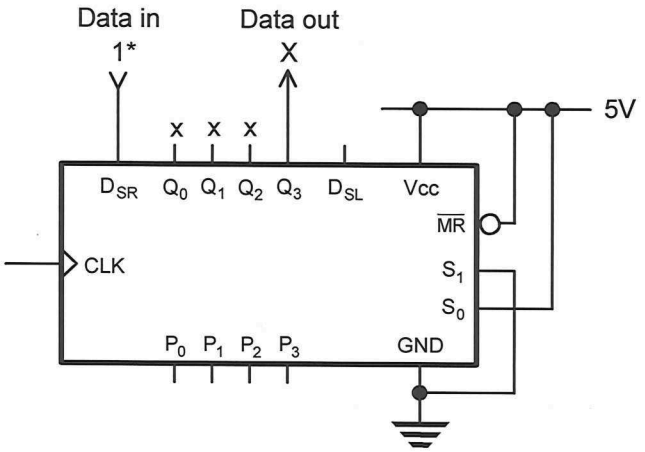


Figure 11

The first clock pulse triggers the device and the data on the D_{SR} input moves to Q_0 (see Figure 12). At the same time, the data on the other outputs shift right one place but they're still meaningless at this point in the circuit's operation.

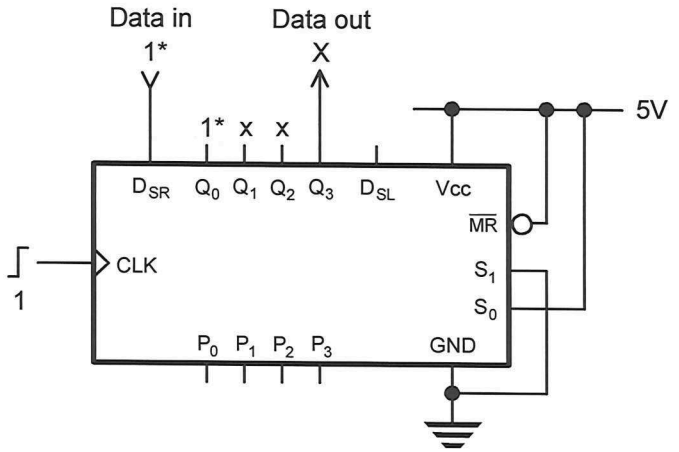


Figure 12

The next bit in the serial data flow (a logic-0 in this example) is applied to the data input and the device is clocked a second time shifting all data another place to the right (see Figure 13).

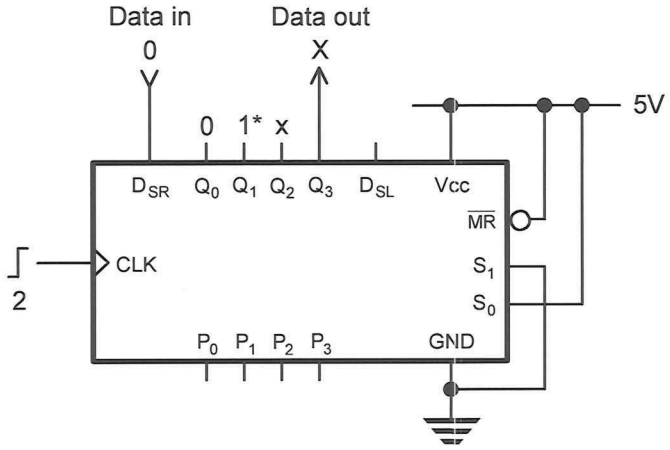


Figure 13

The next bit in the serial data flow (a logic-0 again) is applied to the data input and the device is clocked a third time shifting all data another place to the right (see Figure 14).

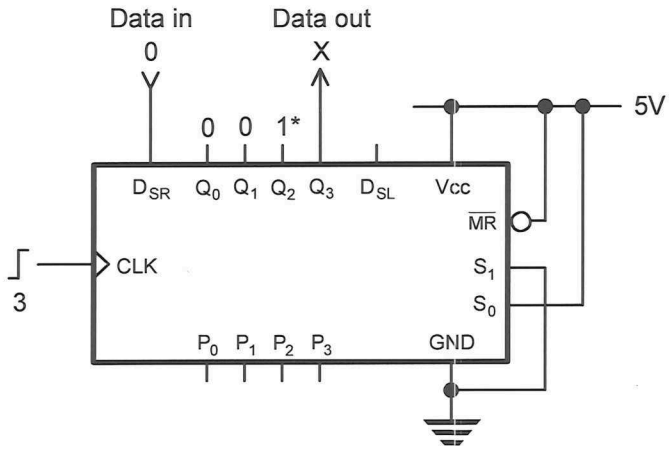


Figure 14

The next bit in the serial data flow (a logic-1 this time) is applied to the data input and the device is clocked a fourth time shifting all data another place to the right (see Figure 15).

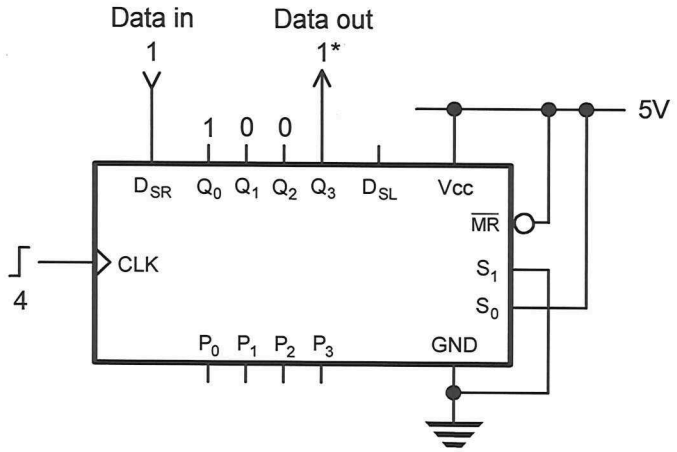


Figure 15

The next bit in the serial data flow (a logic-0 this time) is applied to the data input and the device is clocked a fifth time shifting all data another place to the right (see Figure 16).

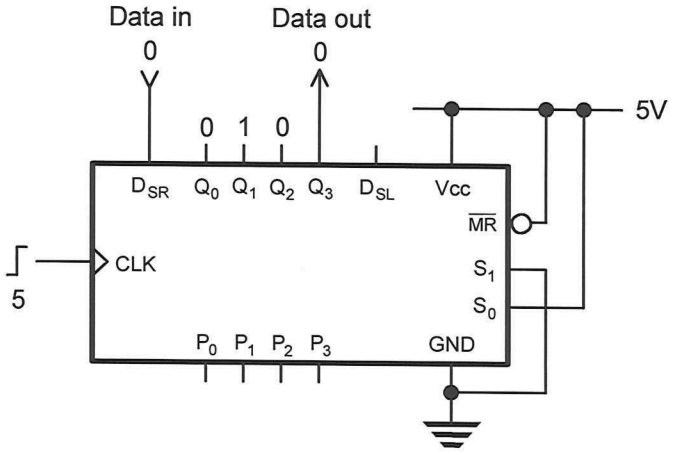


Figure 16

From the serial movement of data shown in Figures 11 to 16 we can make two important observations about serial in, serial out shift registers.

First, once a data bit reaches the serial data output, it is lost on the next clock pulse that triggers the device. This can be seen if you look at the Q-outputs in Figures 12 to 15. The first data bit (the logic-1 with an asterisk next to it) is stored in the register and moves through it from one output to the next each time the device is clocked. But in Figure 16 that particular logic-1 is gone. (Note: There's another logic-1 in the shift register but this is the fourth data bit.)

Second, this is a 4-bit shift register and it takes four clock pulses for the data to travel from the serial input to the serial output. This relationship between the number of the shift register's bits and the number of clock pulse needed for data to travel from the input to the output always applies to serial in, serial out shift registers. So a 10-bit serial in, serial out shift register for example requires ten clock pulses for data to travel from the input to the output.

Other serial in, serial out shift registers using the 74194

If a shift register with fewer than 4-bits is needed, the 74194 can still be used. The circuit is basically the same as the circuit in Figure 11 except one of the other Q-outputs is used for the serial data output instead of Q_3. Figure 17 below shows an example of the 74194 operated as a 3-bit serial in, serial out shift register.

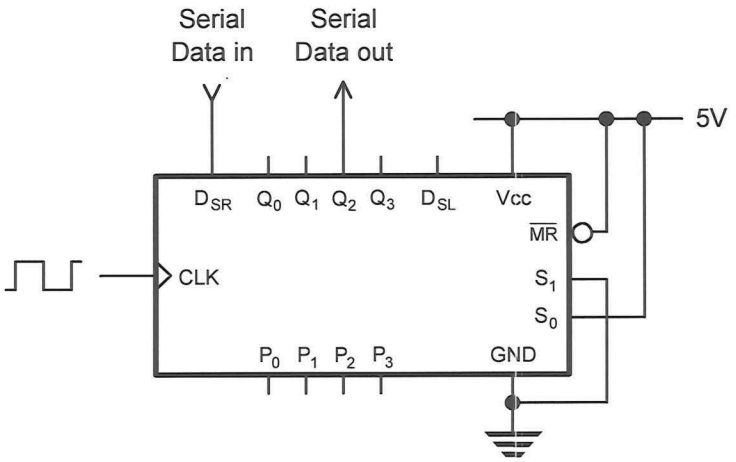


Figure 17 *The 74194 configured to operate as a 3-bit serial in, serial out shift (right) register*

If a shift register with more than four bits is needed, two or more 74194s can be cascaded to provide the necessary number of bits. The ICs are wired in the same way as each other with their clocks connected together. The Q_3 output of each IC is connected to the D_{SR} input of the next IC. Figure 18 is an example of two 74194s used to make an 8-bit serial in, serial out shift register.

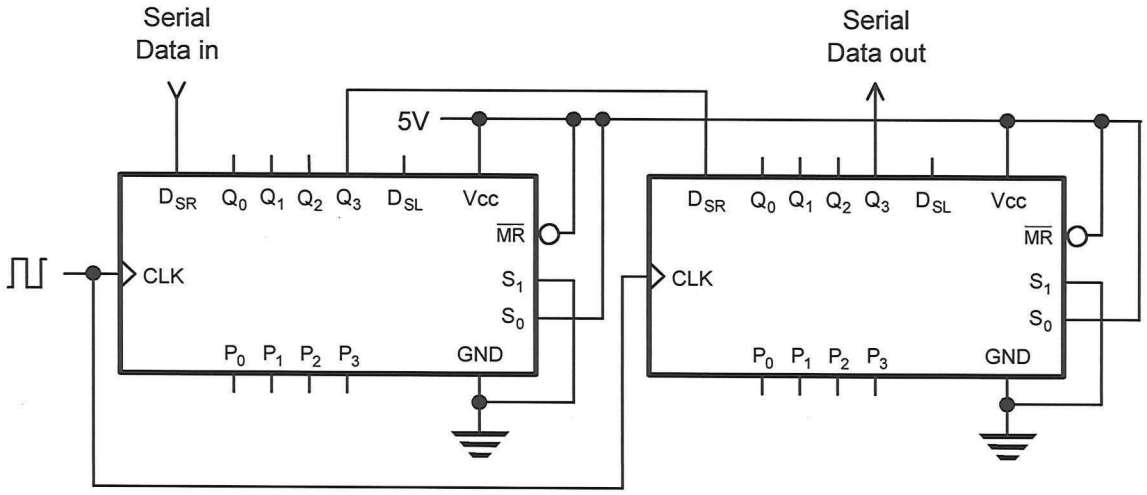


Figure 18 *Two 74194s cascaded to make an 8-bit serial in, serial out shift (right) register*

If a 4-bit shift **left** register is needed, the 74194 is wired as shown in Figure 19 below.

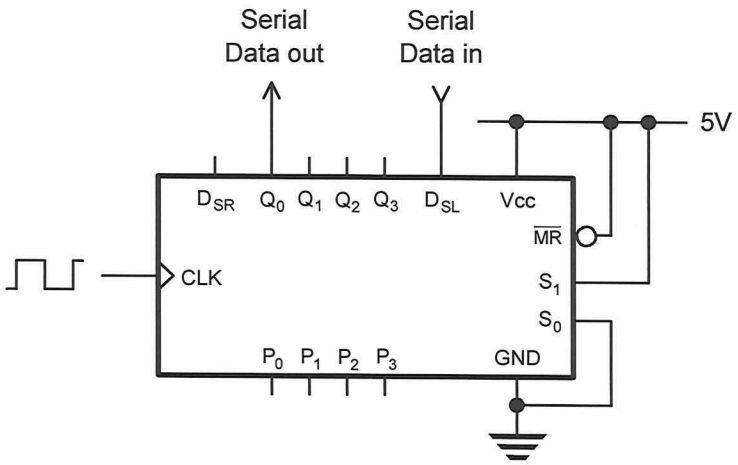


Figure 19 *The 74194 configured to operate as a 4-bit serial in, serial out shift (left) register*

Notice that the serial data into the circuit now connects to the D_{SL} input and the serial data output is taken from Q_0 instead of Q_3. Also, the select inputs are now wired so that S_0 is logic-0 and S_1 is logic-1 instead of the other way round for the shift right register.

Applications

Although the serial in, serial out shift register doesn't seem to do much, it's actually a fairly useful circuit. For example, it can be used to delay data. Data might need delaying if it arrives a point in a circuit before it can be processed. To solve that problem, the data can be passed through a serial in, serial out shift register on its way so that it arrives at circuit at the right time. Additional circuitry could be used to change which of the register's outputs the serial data is taken from providing for a variable delay. Additional circuitry could also be used so that, once the shift register is loaded with a value, its output is connected back to the register's input to recycle the value through the register and delay it for relatively long periods without needing extra flip flops.

Another application for serial in, serial out shift registers is for multiplying and dividing numbers. Suppose you enter the number 0100 (4_{10}) into a 4-bit serial in, serial out shift register. Once done, shifting all bits left one place (with a zero on the input) gives the number 1000 (8_{10}) and you have multiplied the original value by two. Shifting right divides numbers by two.

Like counters, serial in, serial out shift registers can be used as a frequency divider that can also change the clock's mark-space ratio as well as dividing its frequency down. Another application is as a *PN sequence* generator ("PN" stands for pseudo-noise). Circuit examples for these are not shown.

The 74194 operated as a serial in, parallel out shift register

To obtain a serial in, parallel out shift register using the 74194, the chip is operated in the same mode as for the serial in, serial out shift register but all four outputs are used togethera at once instead of just one. The is shown in Figure 20 below.

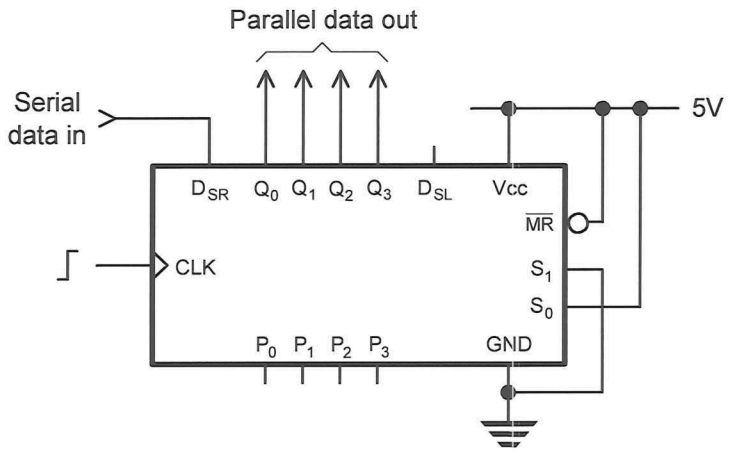


Figure 20 *The minimum wiring requirements to operate the 74194 as a 4-bit serial in, parallel out shift register*

As we have just seen with the operation of the serial in, serial out shift (right) register, data that is entered into the device via the D_{SR} input moves across the outputs one at a time with each clock pulse. So, let's suppose that a 4-bit serial data word like 1001 is entered. After the fourth clock pulse the bits that make up the serial data word occur across the device's four outputs as shown in Figure 21 below. An asterisk denotes the first bit to enter the shift register.

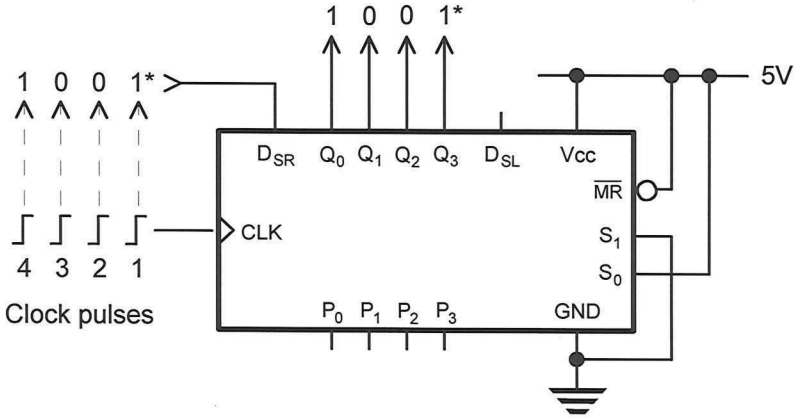


Figure 21 *The outputs after four clock pulses when the data word 1001 is entered in serial form*

Reading the device's four outputs at the same time after the fourth clock pulse (but before the fifth clock pulse) gives us the 4-bit data word in parallel form instead of serial form.

Notice the relationship between the number of bits in the data word and the number of clock pulses it takes before the data word occurs in parallel form. They are always the same. So, an 8-bit serial in, parallel out shift register converts an 8-bit serial data word into an 8-bit parallel data word in eight clock pulses.

Applications

A very useful application for the serial in, parallel out shift register is serial-to-parallel conversion. Serial-to-parallel conversion is needed wherever data is transferred between systems in serial form but is used within the receiving system in parallel form. For example, data sent in serial form to a printer needs to be converted back into parallel form by the printer for use.

The 74194 operated as a parallel in, parallel out shift register

Figure 22 below shows the minimum wiring requirements to operate the 74194 as a 4-bit parallel in, parallel out shift register.

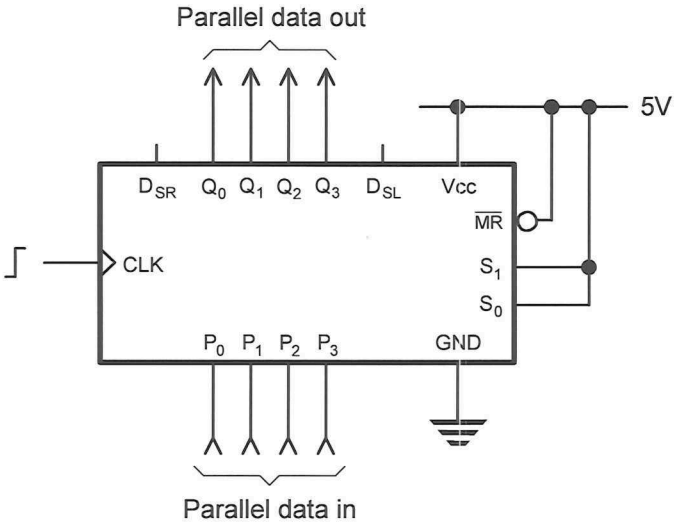


Figure 22 *The minimum wiring requirements to operate the 74194 as a 4-bit parallel in, parallel out shift register*

The operation of this circuit is very straight forward. When the clock signal triggers the device, the logic levels on the P-inputs are transferred almost immediately to the Q-outputs. This is demonstrated in Figures 23 and 24 as the 4-bit data word 1001 is applied to the parallel input then the device is clocked.

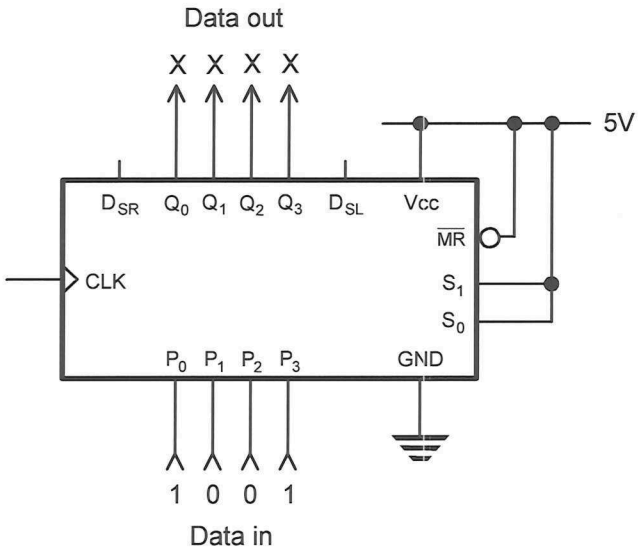


Figure 23 *Initially, the 4-bit data word 1001 is placed on the P-inputs but the device has not yet been clocked so all of the outputs have not been determined by the circuit and so are meaningless*

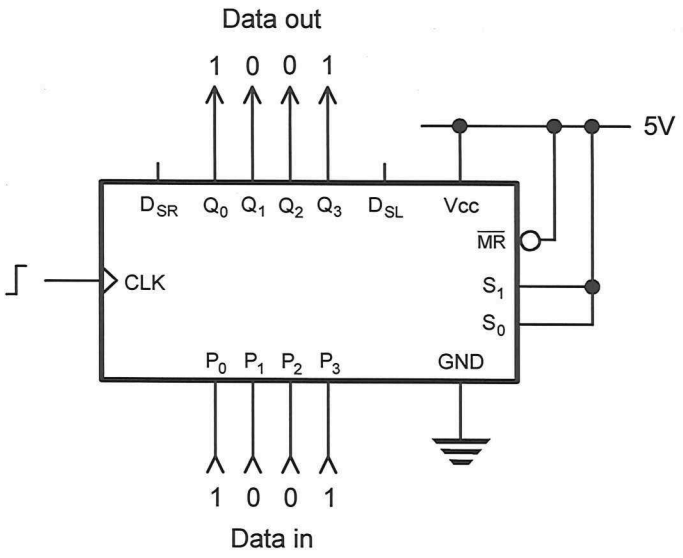


Figure 24 *A positive edge on the clock input triggers the device and the data on the inputs is copied to the corresponding outputs*

From the parallel movement of data shown in Figures 23 and 24 we can see that it takes only one clock pulse to transfer the data from input to output. Importantly, if the data on the parallel inputs changes, the data on the outputs does not change unless the device is clocked again.

Applications

The fact that data entered into a parallel in, parallel out shift register wont change unless the device is clocked again (or the power is turned off then turned back on) means that the register has latched the data. For this reason, the parallel in, parallel out shift register is often called a *data latch*.

Data latches are useful circuits wherever data must be caught and held onto for short periods of time or indefinitely. They are used extensively inside microprocessor-based systems like personal computers and inside microprocessors themselves.

For an example of why data may need to be trapped for short periods of time, consider a 4-bit serial in, parallel out shift register. As discussed a couple of pages ago, it takes four clock pulses to convert the 4-bit serial data word into a 4-bit parallel data word. Importantly, on the fifth clock the data on the outputs changes (as a new conversion begins) and the 4-bit parallel data word from the first conversion is lost. This may be a problem if the circuit using the parallel data has not had time to do something with it before it is lost. To avoid this problem a parallel in, parallel out shift register is used to capture the parallel data word as shown in Figure 25 on the next page.

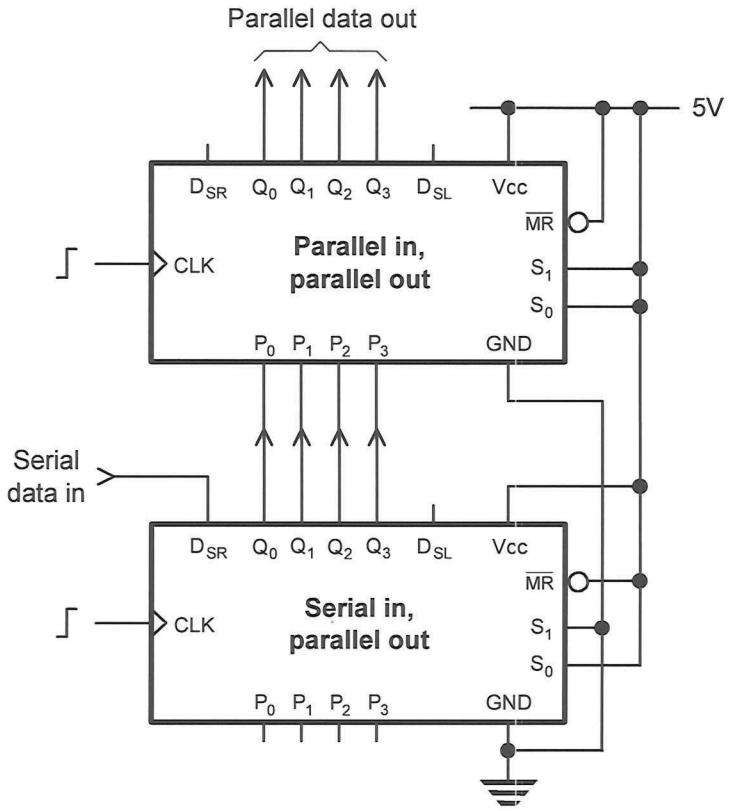


Figure 25 *Serial in, parallel out shift register with parallel data hold using a parallel in, parallel out shift register*

The upper part of the circuit is the 4-bit parallel in, parallel out shift register first shown in Figure 22 on page 5-16. The source of its parallel data is the 4-bit serial in, parallel out shift register first shown in Figure 20 on page 5-14.

To capture and hold the parallel data output of the serial in, parallel out shift register the parallel in, parallel out shift register must be clocked after the fourth clock pulse but before the fifth. After that, the parallel in, parallel out shift register can be allowed to hold onto the parallel data for four more clock pulses when it must be clocked again to capture and hold the parallel data from the next conversion.

Clocking the parallel in, parallel out shift register at precisely the right time might seem like it requires a masterstroke of timing but in fact it is simple using a counter IC like the 74193. A circuit is shown in Figure 26 on the next page. (Figure 26 is provided to give you a feel for how digital components can be combined to implement sub-systems. Neither the circuit, nor its operation are assessed.)

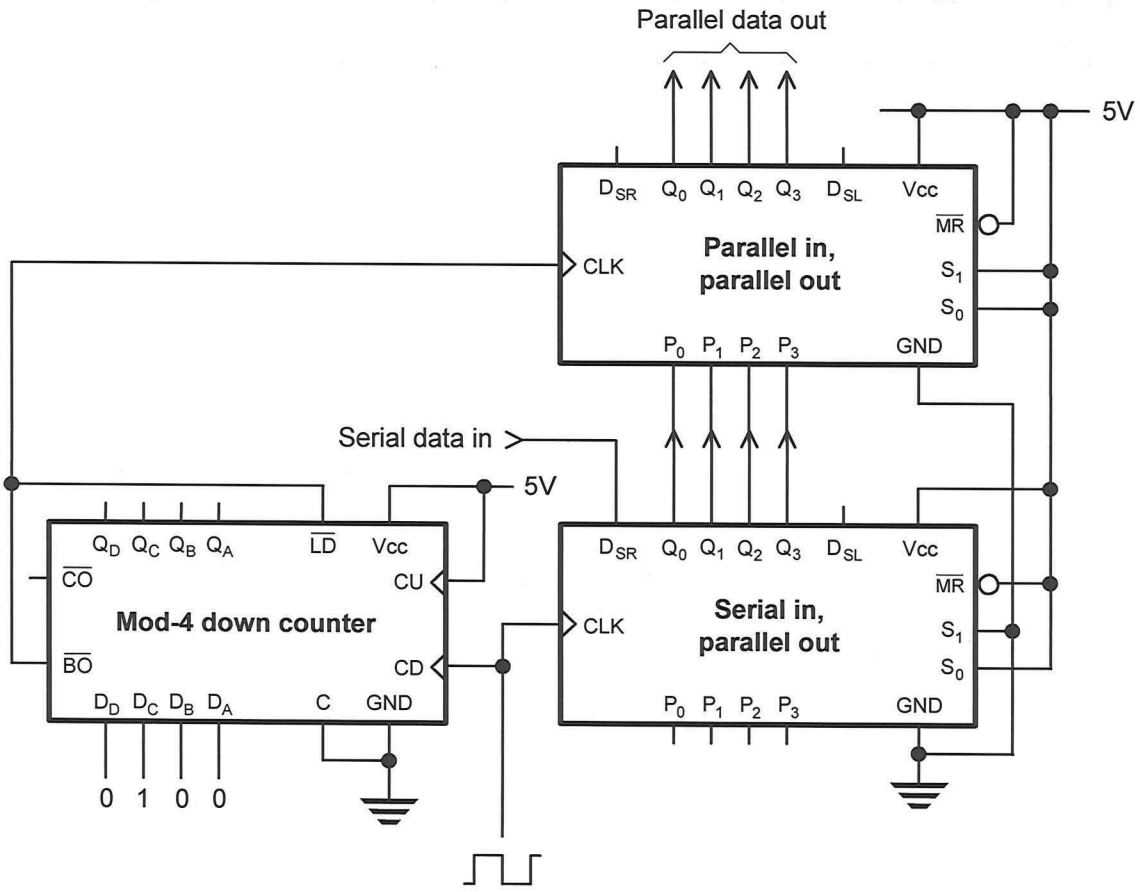


Figure 26 *The circuit in Figure 25 with the clocking of the parallel in, parallel out shift register managed automatically using the 74193 counter IC*

A 74193 is wired as a Mod-4 down counter that counts down from 0100 (4_{10}) to 0001 (1_{10}). When the counter's output is clocked the next time after the output reaches 0001, it changes to 0000 but only briefly because, at the same time, the device's $\overline{BO}$ output goes low. This asserts the counter's $\overline{LD}$ input replacing 0000 on the output with 0100 without having to be clocked (so the count sequence over four clock pulses is 0011, 0010, 0001 back to 0100).

Importantly, the down counter's clock input is connected to the same clock signal that triggers the serial in, parallel out shift register. This means that the counter counts down by one every time the serial in, parallel out shift register shifts data right one place.

Equally importantly, the down counter's $\overline{BO}$ is connected to the positive edge-triggered clock input of the parallel in, parallel out shift register at the top. Recall that his is an output that rests on logic-1 for most of the count sequence. When the number on the counter's output is 0000 it goes to logic-0 and returns to a logic-1 when the counter overflows or is preset with a non-zero number.

Putting this this information together, events around the system are as follows. At the beginning of a new conversion, the down counter's output is 0100 and the first bit of the 4-bit data word to be converted is on the serial in, parallel out shift register's D_{SR} input. After four clock pulses, the 4-bit data word is converted to parallel and available on the serial in, parallel out shift register's output. At the same time the this happens counter's output is 0000 (though, only momentarily - keep reading).

When the down counter's output changes to 0000, $\overline{BO}$ goes low. This asserts the down counter's $\overline{LD}$ input causing its output to change back to 0100. Importantly, this causes $\overline{BO}$ to return to a logic-1. This transition from a logic-0 to a logic-1on $\overline{BO}$ clocks the parallel in, parallel out shift register causing the converted 4-bit data word on the serial in, parallel out shift register's output to be transferred to its own output where it's latched and is available for four more clock pulses before it is replaced by the result of the next conversion.

The 74194 operated as a parallel in, serial out shift register

Operating the 74194 as a parallel in, serial out shift register requires the device to be operated in two modes. The parallel load mode is used first followed by either the shift left or shift right modes (depending on the shift direction required).

Figure 27 below shows the 74194 wired to operate as a parallel in, serial out shift register. The select inputs connect to other circuitry not shown here that puts the device into the required mode at the right time.

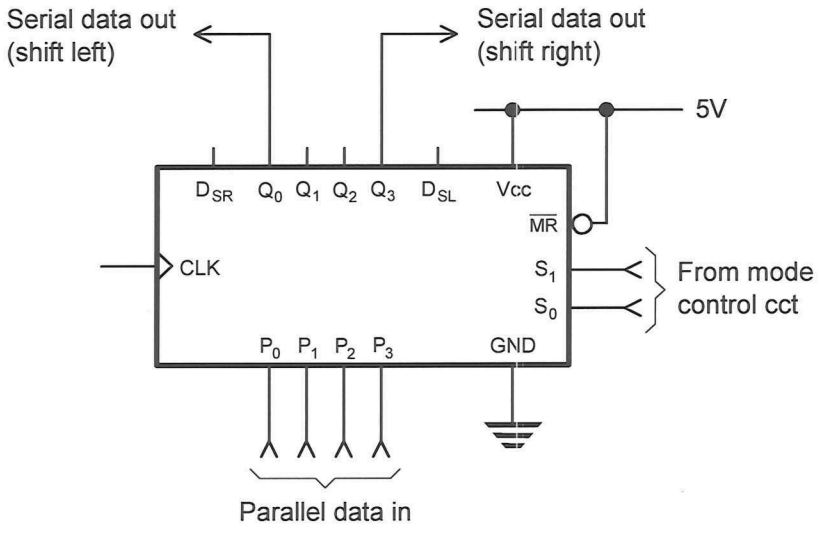


Figure 27 *The wiring requirements (not including mode control circuitry) to operate the 74194 as a parallel in, serial out shift register*

To demonstrate the operation of the 74194 as a parallel in, serial out shift register, Figures 28 to 33 show the 4-bit parallel data word 1001 being loaded then shifted right.

Initially, the 4-bit parallel data word is applied to the parallel inputs and the select inputs are made logic-1 to put the 74194 onto parallel load mode.

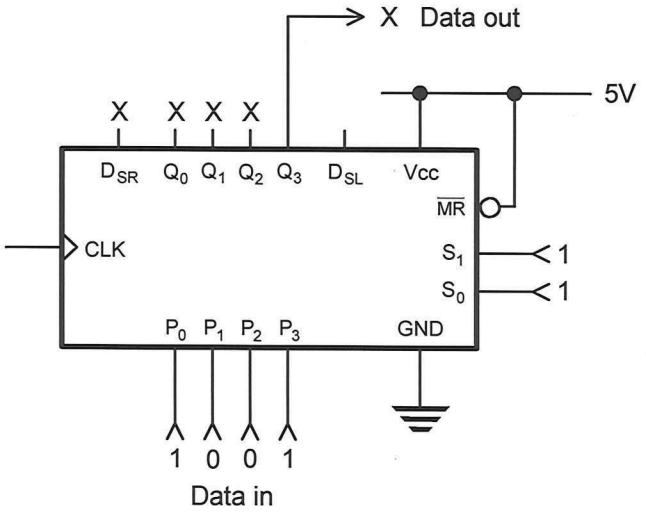


Figure 28

The device is then clocked and the 4-bit parallel data word moves to the outputs and the first of the four bits is available to be read out serially.

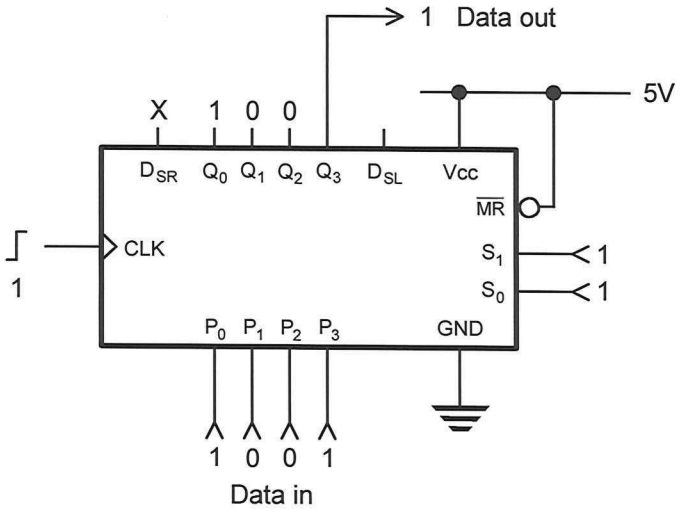


Figure 29

The 74194 is then put into shift mode (shift right in this example) making the data on the parallel inputs meaningless (and can now change without affecting the device's outputs).

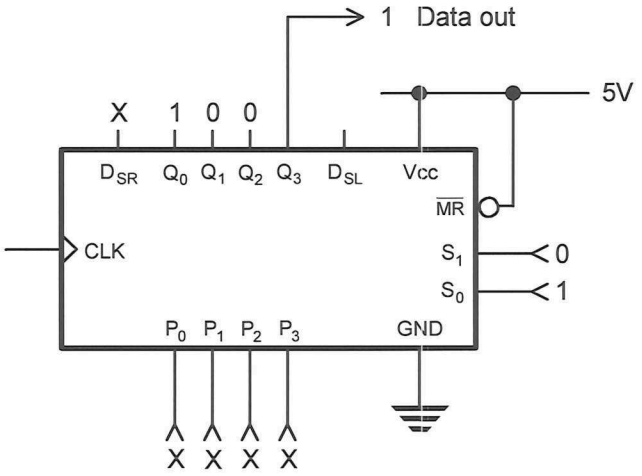


Figure 30

The device is then clocked a second time shifting all bits to the right making the second bit in the 4-bit data parallel data word available to be read out serially (see Figure 31). Importantly, notice that the logic level on the D_{SR} input is copied to the Q_0 output. For the purposes of our explanation here we've not worried about what the logic level is but have distinguished it with an X.

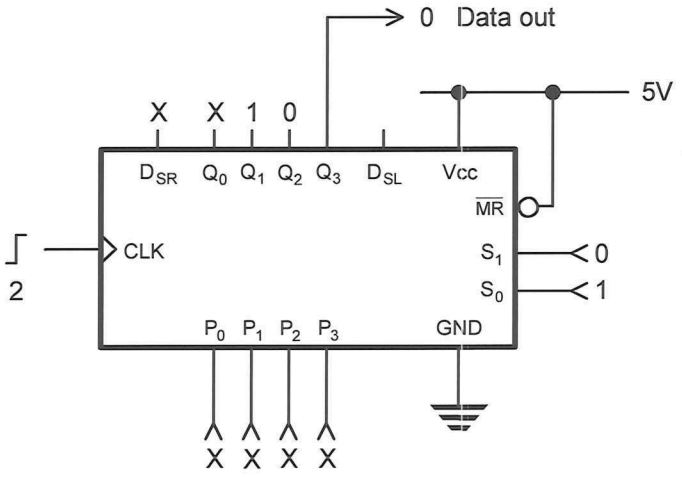


Figure 31

The device is then clocked a third time making the third bit in the 4-bit data parallel data word available to be read out serially (see Figure 32).

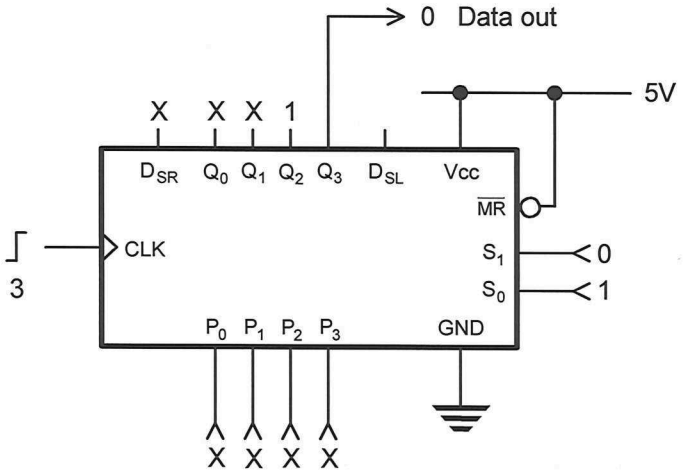


Figure 32

Finally, the device is clocked a fourth time making the fourth bit in the 4-bit data parallel data word available to be read out serially (see Figure 33).

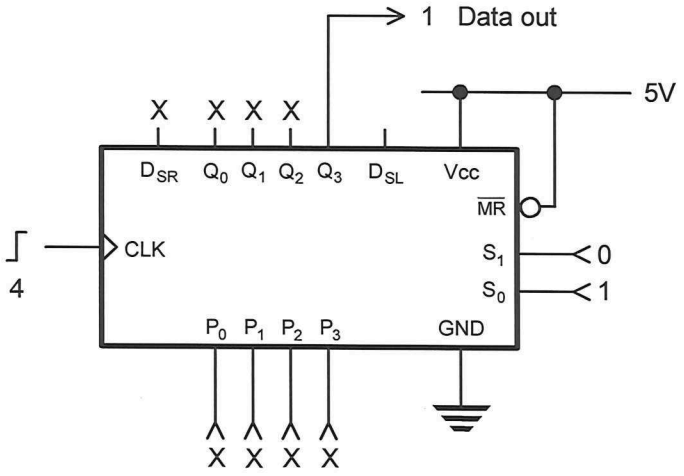


Figure 33

The fifth clock pulse marks the beginning of a new cycle. For the serial data out to remain meaningful, another parallel data word must be applied to the parallel inputs and the device must be put into parallel load mode after the fourth clock pulse but before the fifth.

Control of the select inputs (S_0 and S_1) to make sure that they're the right logic level at the right time can be managed using an IC counter like the 74193. A circuit is shown in Figure 34 below. (Again, Figure 34 is provided to give you a feel for how digital components can be combined to implement sub-systems. Neither the circuit, nor its operation are assessed.)

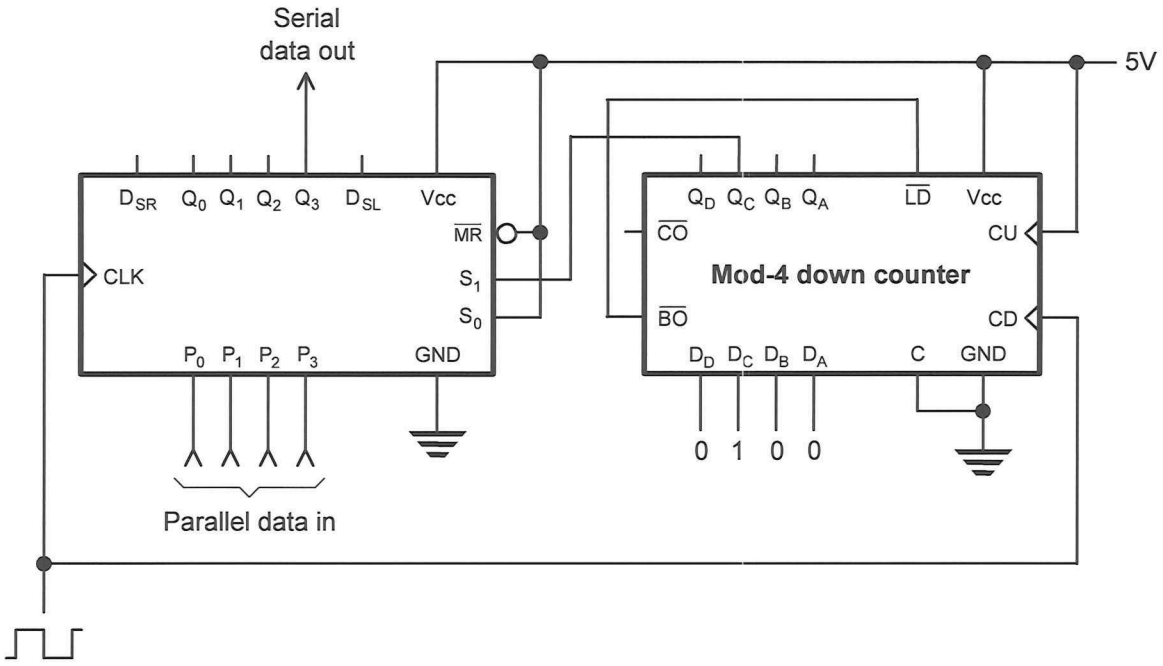


Figure 34 *The circuit of a parallel in, serial out shift register with the mode control circuitry being managed automatically using the 74193 counter IC*

The S_0 select input is hard-wired to a logic-1 as it needs to be this logic level for both parallel load and shift right modes. The S_1 select input is connected to the Q_C output of the 74193 wired as a Mod-4 down counter.

On the clock that causes the counter to overflow, the number 0100 is loaded onto the counter's output. This means that the Q_C and S_0 are logic-1 and so the 74194 is in parallel load mode. That being the case, next clock pulse presets the shift register's outputs with a parallel data word. At the same time, the counter counts down one to 0011. Importantly, this means that Q_C and S_0 are logic-0 putting the 74194 into shift right mode. The next three clock pulses subsequently shift out the data and cause the counter to overflow readying the circuit for the next conversion.

Applications

The most important application for the parallel in, serial out shift register is parallel-to-serial conversion. Parallel-to-serial conversion is needed wherever data is used within two (or more) systems in parallel form but must be transferred between them in serial form. For example, a personal computer uses data in parallel form but the data link between a computer and its printer is usually serial (assuming the data link is RS232C or USB).

Skill practice 5

Practise testing and measuring sequential logic shift register ICs

This exercise is practise for the sorts of skills you may be required to perform in a practical test. Remember, in any practical tests you will be working alone so make sure that you can perform all the steps. It should take you about 1¼ hours to complete this exercise.

Equipment

- Digital trainer
- logic probe
- 74LS194
- hook-up wire (terminated)
- IC extractor

Remember:

Follow TAFE NSW WHS guidance at all times!

 Copyright: Not to be reproduced without permission

Work tasks

1. Read your WHS responsibilities at the top of the form below. Then conduct a WHS risk assessment and record your findings in the space provided.

Responsibilities of students under the Model WHS Act: s28

- Take reasonable care for your own health and safety by working safely at all times
- Take reasonable care to ensure that your acts or omissions don't put the health and safety of others at risk
- Follow all TAFE NSW WHS guidance and comply with all reasonable instructions from TAFE NSW staff to assist them in complying with the TAFE NSW WHS requirements
- In addition to the above, you must:
  - use and maintain machinery, tools and all other equipment properly and safely
  - ensure that your work area is free of hazards
  - notify a TAFE NSW staff member of actual or potential hazards
  - wear/use prescribed safety equipment
  - take notice of any safety signs and adhere to their instructions

Risks involved in this activity include:	**Control measures:**
■ Trip hazards (eg students bags) ■ Objects dropped on feet (while equipment is taken to and from benches).	■ Move bags and other objects from walkways ■ Plan lifting of equipment
Others: _____ _____ _____	Other: _____ _____ _____

My signature here indicates that I have read and understand my responsibilities under the Model WHS Act s28 (detailed above). I have also conducted a risk assessment before undertaking this activity and have identified measures to control these risks and have implemented them.

Signature: _____ Date: _____

2. Gather the equipment needed for this exercise.

3. Read the following points and make sure that you perform them when wiring all circuits:

   - Write the IC pin numbers of the devices you work with on the circuit that you're instructed to wire.
   - Handle all ICs carefully - remember ESD can damage them.
   - Ensure the power supply is connected to all ICs in the circuit.
   - When plugging hook-up wire into the breadboard to make connections to the IC's pins, use holes that are furthest away from the pin.

4. Wire the circuit of Figure 1 on the digital trainer. Pinout data for the 74LS194 can be found in Appendix 6.

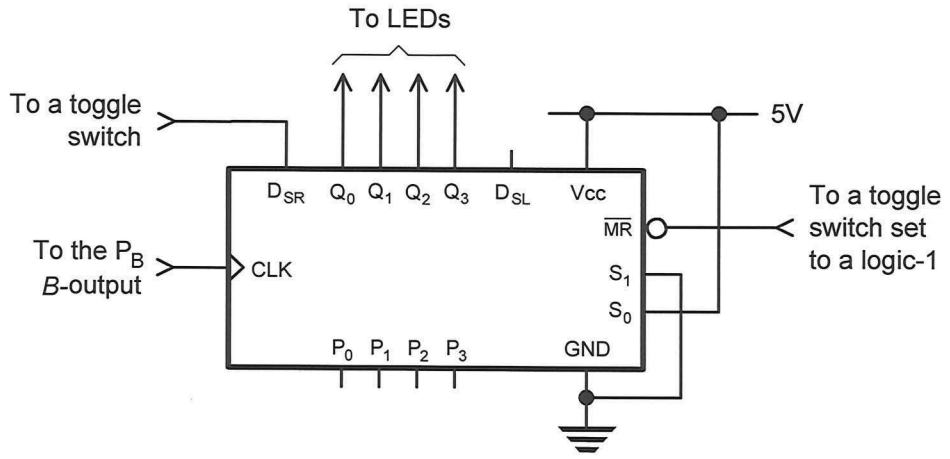


Figure 1

5. Turn on the trainer and clear the shift register by momentarily making the toggle switch connected to the $\overline{MR}$ input a logic-0.

 Copyright: Not to be reproduced without permission

6. Use the toggle switch to make the data on the D_{SR} input a logic-1.

7. Use the pulser to trigger the shift register once. Record the data on the register's outputs in Table 1 below.

8. Complete the table by making the D_{SR} input the logic level listed in each row of the *Data in* column and clocking the device once each time.

Table 1

Clock pulse	Data in	Outputs			
		Q_0	Q_1	Q_2	Q_3
-	-	0	0	0	0
1	1				
2	0				
3	0				
4	0				
5	0				
6	1				
7	0				
8	1				
9	1				
10	0				
11	0				
12	0				
13	0				

Question 1
What happens to the data on the D_{SR} input each time the device is clocked?

Question 2
What happens to the data on the Q-outputs each time the device is clocked?

Question 3
Which output or outputs provide a copy of the input data after four clock pulses?

- ☐ Q_0
- ☐ Q_1
- ☐ Q_2
- ☐ Q_3
- ☐ All four of them

Question 4
What type of shift register is the circuit operating as if data is entered via the D_{SR} input and only taken out of Q_3?

- ☐ Serial in, serial out
- ☐ Serial in, parallel out
- ☐ Parallel in, serial out
- ☐ Parallel in, parallel out

Question 5
Which output or outputs provide a 4-bit parallel data word out of the shift register after four clock pulses?

- ☐ Q_0
- ☐ Q_1
- ☐ Q_2
- ☐ Q_3
- ☐ All four of them

Question 6
What type of shift register is the circuit operating as if data entered via the D_{SR} input and taken out of all four outputs simultaneously?

- ☐ Serial in, serial out
- ☐ Serial in, parallel out
- ☐ Parallel in, serial out
- ☐ Parallel in, parallel out

 Copyright: Not to be reproduced without permission

The teacher needs to check your work at this point...

9. Turn off the trainer and wire the circuit of Figure 2 below. Hard-wire the D_{SR} and parallel load inputs to the logic levels shown.

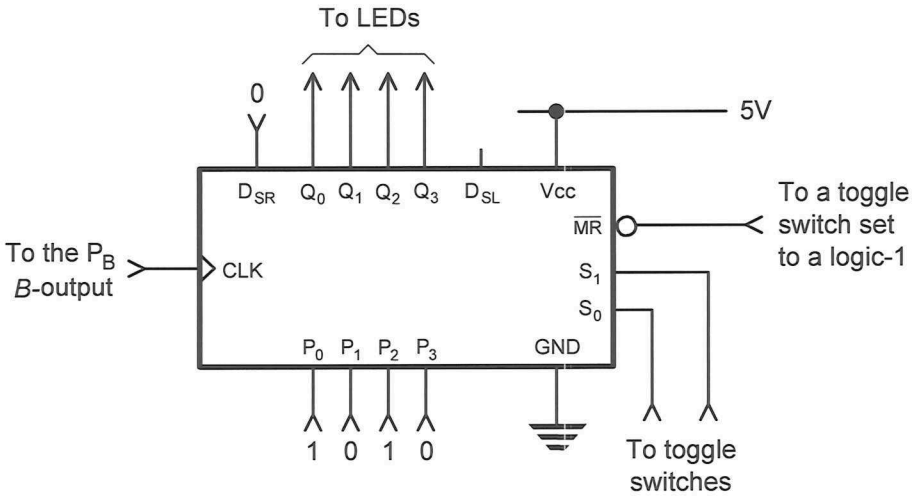


Figure 2

10. Turn on the trainer and clear the shift register by momentarily making the toggle switch connected to the input a logic-0.

11. Use the toggle switches to make both of the select inputs (S_0 and S_1) logic-1.

12. Use the pulser to trigger the shift register once. Record the data on the register's outputs in Table 2 below.

13. Use the toggle switches to make S_0 a logic-1 and S_1 a logic-0.

14. Trigger the shift register again and record the logic levels on the outputs.

15. Repeat step 14 until the table is complete.

Table 2

Clock pulse	Outputs			
	Q_0	Q_1	Q_2	Q_3
-	0	0	0	0
1				
2				
3				
4				

Question 7
Which output or outputs provide parallel data out of the shift register after the first clock pulses?

- ☐ Q_0
- ☐ Q_1
- ☐ Q_2
- ☐ Q_3
- ☐ All four of them

Question 8
What type of shift register is the circuit operating as if data is entered via the P-inputs and taken out of all four outputs simultaneously?

- ☐ Serial in, serial out
- ☐ Serial in, parallel out
- ☐ Parallel in, serial out
- ☐ Parallel in, parallel out

 Copyright: Not to be reproduced without permission

Question 9
What type of shift register is the circuit operating as if data is entered via the P-inputs and only taken out of Q_3?

- ☐ Serial in, serial out
- ☐ Serial in, parallel out
- ☐ Parallel in, serial out
- ☐ Parallel in, parallel out

The teacher needs to check your work at this point...

Review questions

Answer these questions to check your understanding of what you have learnt from this chapter. Doing this will also help to prepare you for the assessments.

1. What is the difference between serial and parallel data transfer?

2. What is the advantage of parallel data transfer over serial?

3. What is the advantage of serial data transfer over parallel?

4. Shift registers are made out of cascaded

   ☐ NOR RS flip flops.
   ☐ NAND RS flip flops.
   ☐ D-type flip flops.
   ☐ toggling flip flops.

5. List three applications for a serial in, serial out shift register.

Questions 6 to 8 refer the block diagram of a device wired as an 8-bit serial in, serial out shift (right) register in Figure 1

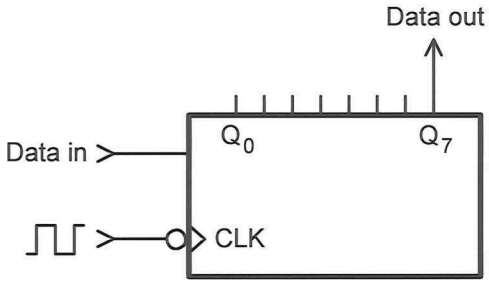


Figure 1

6. The clock input is

   ☐ active high.
   ☐ active low.
   ☐ positive edge-triggered.
   ☐ negative edge-triggered.

7. How many clock pulses does it take for serial data entered into the device to appear at the output?

   _____

8. On what output would input data appear after only 5 clock pulses?

   _____

9. Give an application for a serial in, parallel out shift register.

Questions 10 to 12 refer the block diagram of a device wired as an 8-bit serial in, parallel out shift (right) register in Figure 2

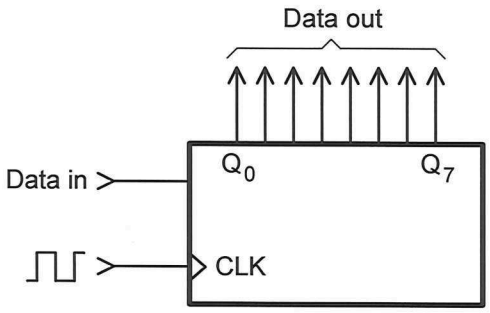


Figure 2

10. The clock input is

- ☐ active high.
- ☐ active low.
- ☐ positive edge-triggered.
- ☐ negative edge-triggered.

11. How many clock pulses does it take for a serial data word entered into the device to appear as a parallel data word across the outputs?

12. What is the output data after the device is clocked if, before being clocked, the device's output data is 10001001 and the data input is logic-1?

 Copyright: Not to be reproduced without permission

13. Give an application for a parallel in, parallel out shift register.

Questions 14 to 16 refer the block diagram of an 8-bit parallel in, parallel out shift register in Figure 3

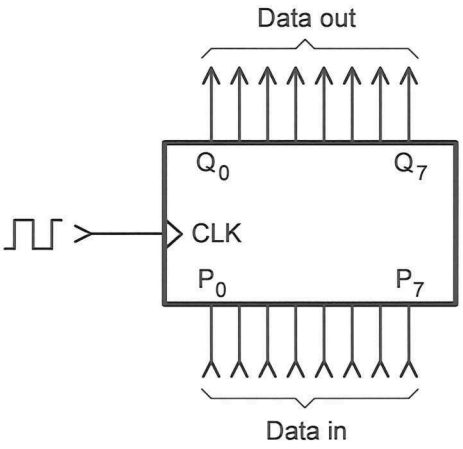


Figure 3

14. What is another name for this circuit?

15. How many clock pulses does it take for a parallel data word entered into the device to appear as a parallel data word on the output?

16. What is the output data after the device is clocked if, before being clocked, the device's output data is 01101110 and the input is 00001100?

17. Give an application for a parallel in, serial out shift register.

_____

Questions 18 to 20 refer to the 74LS164 IC. Data for the device can be found in Appendix 6

18. The clear input is

- ☐ positive edge-triggered.
- ☐ negative edge-triggered.
- ☐ active high enabled.
- ☐ active low enabled.

19. The clock input is

- ☐ positive edge-triggered.
- ☐ negative edge-triggered.
- ☐ active high enabled.
- ☐ active low enabled.

20. The device can be used as a

- ☐ data latch.
- ☐ serial-to-parallel converter.
- ☐ parallel-to-serial converter.
- ☐ all of the above.

21. The device converts

- ☐ 4 bit words.
- ☐ 8 bit words.
- ☐ 10 bit words.
- ☐ 12 bit words.

Copyright: Not to be reproduced without permission

Student notes

Section 6 Encoders and decoders

Purpose To develop your understanding of encoders and decoders and to develop your ability to interpret the manufacturers' data sheets for common commercially available encoder and decoder ICs.

Objectives At the end of this section you should be able to:

- Explain the basic operation of an encoder and a decoder (at the block diagram level only)
- Interpret the truth table of encoders and decoders
- Explain the difference between an encoder and a priority encoder
- Give typical applications for encoders and decoders
- Interpret manufacturers' data for common commercially available encoder and decoder ICs
- Use a logic probe to test and measure encoders and decoders

Introduction

In this section you'll learn about the operation of another set of useful logic devices called *encoders* and *decoders*. You'll look at applications for these devices and examples of common commercially available encoder and decoder ICs.

Encoders

An encoder is a device that has two or more inputs that are numbered using decimal (typically from zero or one up). When one of the inputs is asserted, the encoder outputs a binary number equivalent to the input's number.

The operation of an encoder might be better understood if we consider a simple example. Figure 1 below shows the block diagram of an *4-line-to-2-line* encoder. The logic circuit that implements this device is not shown here. However, the circuit and a brief explanation are provided in Appendix 2 for your interest.

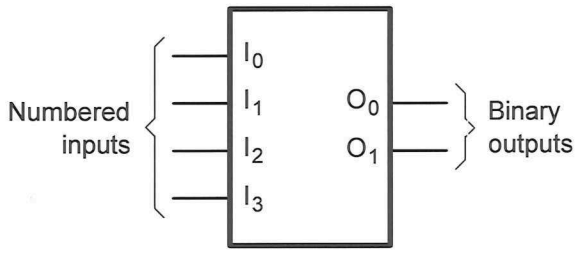


Figure 1 *The block diagram of a 4-line-to-2-line encoder*

As you can see, the device has two outputs which are labelled O_0 and O_1. Together the outputs allow the device to output a 2-bit binary number (eg 00, 01, 10 or 11).

The device also has four inputs that are numbered I_0 to I_3. The block diagram shows that the inputs on this example of an encoder are active high. That is, the logic level on the inputs should normally be logic-0 and applying a logic-1 to an input asserts it. Encoders with active low inputs are also available.

The device's operation is as follows. Whenever an input is asserted the encoder outputs a binary number equivalent to the input's decimal number. For example, if Input-0 (I_0) is asserted the device outputs the number 00, if Input-1 (I_1) is asserted the device outputs the number 01 and so on. This is shown in Figures 2a to 2d on the next page.

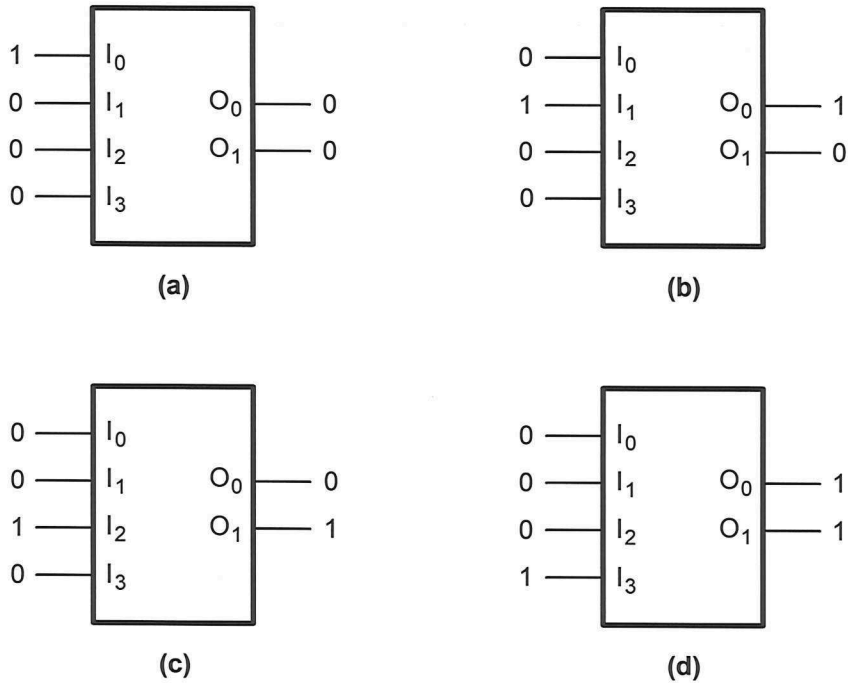


Figure 2 *a) When Input-0 is asserted the device outputs the binary number 00; b) When Input-1 is asserted the device outputs the binary number 01; c) When Input-2 is asserted the device outputs the binary number 10; and d) When Input-3 is asserted the device outputs the binary number 11*

The device's operation gives us a clue why it's called an encoder. As Figures 2a to 2d show, the encoder converts or *encodes* digital information from a non-binary form into binary. That is, 0001 on the input gets converted to 00 at the output; 0010 gets converted to 01; 0100 gets converted to 10; and 1000 gets converted to 11.

This might leave you to ask an important question about encoders: What do they do when none of its inputs are asserted? Asking the same question another way, what code appears on the output when 0000 is applied to the input?

This problem is handled by the manufacturers of commercially available encoder ICs in two similar but different ways. One is to design the device so that, when none of the inputs are asserted, it simply outputs the same binary number as when Input-0 is asserted. That is, both 0000 and 0001 on the input cause the device to output 00.

However, this design solution creates a small problem of its own. As asserting Input-0 is the same as not asserting Input-0, then that input isn't needed! So, the second design solution for encoder ICs is to leave Input-0 off and assume it to be asserted when none of the other inputs are. This type of 4-line-to-2-line encoder has only three inputs and its block diagram is shown in Figure 3 below.

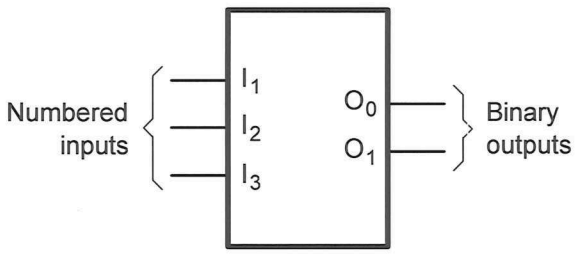


Figure 3 *The block diagram of a 4-line-to-2-line encoder with Input-0 left off*

The operation of this device is shown in Figures 4a to 4d below.

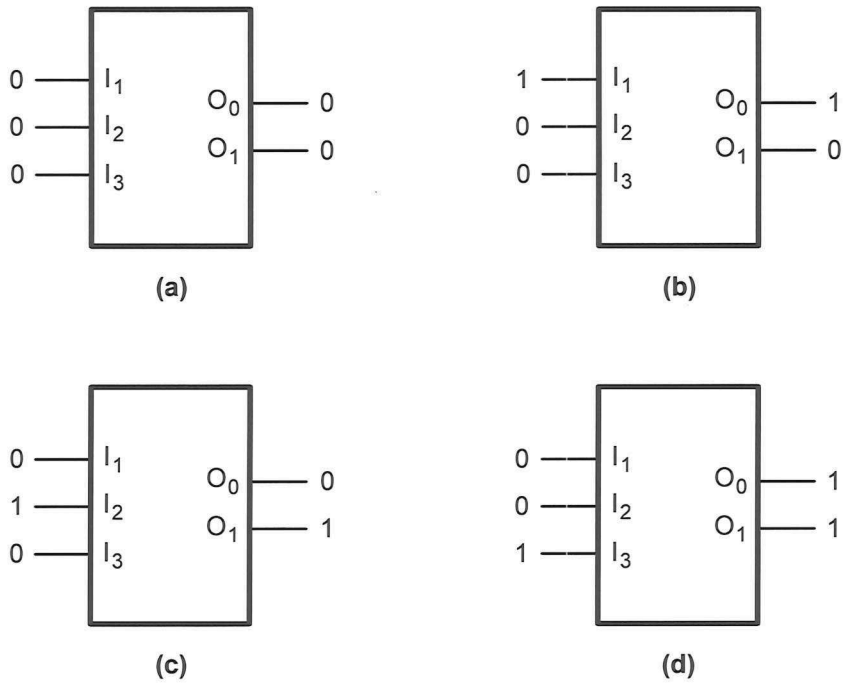


Figure 4 *The relationship between inputs and outputs for the device in Figure 3*

Applications

Encoders find use in digital control systems for converting attributes of a physical system being monitored to binary numbers for ease of processing.

That's a little vague so, for a concrete example, think of a ceiling fan that can be turned off or can be turned on at one of three speeds. Old-school versions of this sort of thing (so, not digitally controlled) use a four-position rotary switch that allows the operator to disconnect the power to turn the fan off, or switch in one of three different value resistors to turn the fan on but at different speeds.

However, using different values of resistance is not how digital control systems work to control things like rotational speed and other methods are used (for example, pulse width modulation). So, digitally-controlled ceiling fan might have four push-buttons instead. Briefly pushing the button marked 0 turns the fan off. And briefly pushing one of the other buttons turns the fan on at different speeds with 1 being the slowest and 3 being the fastest.

For such an arrangement, there needs to be a way of turning the button-press into information involving 0s and 1s that the digital system can do something with and using binary numbers make sense. This is where an encoder can be used and a simple circuit is shown in Figure 5 below.

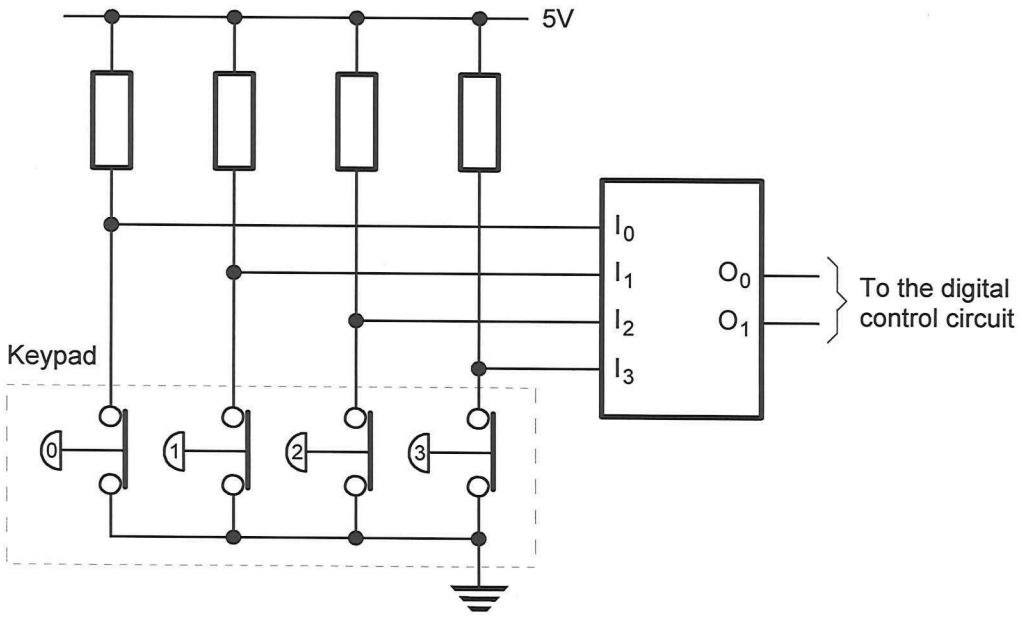


Figure 5 *A simple 4-option touch button encoder*

As you can see, the circuit consists of a 4-line-to-2-line encoder with active high inputs. Each input is connected to both a pull-up resistor to supply and a normally-closed push-button connected to common. Being normally-closed the buttons ensure that the encoder's inputs are all connected to common by default. This means that the default logic level on all of the encoder's inputs is logic-0.

As the circuit shows, the marked number for each push-button is the same as the number of the encoder's input that it's connected to. When a button is pushed, the resulting open-circuit removes the short to ground for that button's input. This allows that input's resistor to pull it up to logic-1. As we know from the previous discussion, a logic-1 to an encoder with active high inputs asserts the input causing the encoder to generate a binary number that corresponds with input's decimal number. The digital circuit connected to the encoder's output is designed to take this information and use it to turn the fan off or turn it on at varying speeds depending on the binary number generated.

Encoder ICs

There's not a lot of variation on the basic encoding idea so there's not many commercially available encoder ICs. Examples include the 74147 and the 74148. To give you a feel for working with encoder ICs we'll look at just one of these devices and discuss its operation in a little detail.

The 74147 encoder

This TTL device is a 10-line-to-4-line priority encoder with active low inputs and outputs. The pinout for the 74147 is shown in Figure 6 below. Notice that the labelling of the inputs is slightly different to the example earlier. There are always variations in this regard between manufacturers' data.

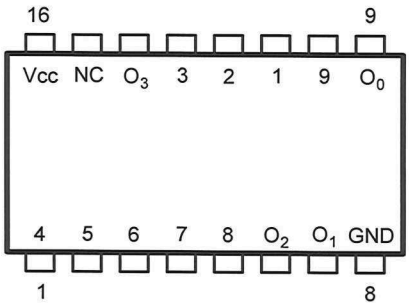


Figure 6 *Pinout data for the 74147*

To help make the device's operation a little easier to understand, it has been redrawn in Figure 7 below in block diagram form with the pins arranged more conveniently.

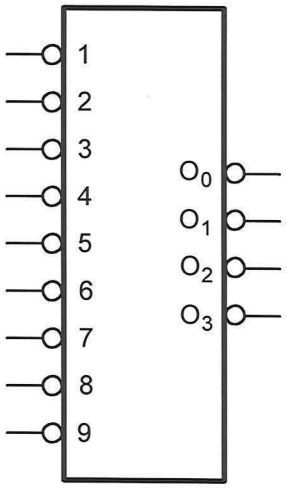


Figure 7 *A block diagram for the 10-line-to-4-line priority encoder inside the 74147*

Unlike most of the digital ICs that we've looked at in other chapters of this workbook, this device has only one mode of operation and so wiring the IC is pretty straight forward and is not shown here. However, the encoder inside the 74147 is a little different to the simple encoder in Figure 1 in three ways and these need to be explained.

First, as the block diagram in Figure 7 shows, the inputs are active low. This means that the default logic level for the inputs is logic-1 and a logic-0 is needed to assert them. This is shown in the device's truth table below (Table 1). Second, you'll notice that the outputs are also active low. So, when none of the inputs are asserted for example, the device outputs the number 1111 instead of 0000. When Input-1 is asserted, the device outputs the number 1110 instead of 0001 and so on. This is also shown in Table 1.

Third and most importantly, the encoder in the 74147 is a *priority* encoder. This means that if two or more inputs are asserted at the same time, the device has been designed to give priority to the highest value input and ignore the lower value inputs. This aspect of the device's operation is highlighted in its truth table by using a "×" for all inputs below the asserted input. The × represents "don't care". That is, it doesn't matter whether the input is logic-0 or logic-1.

Table 1

Inputs									Outputs			
1	2	3	4	5	6	7	8	9	O_3	O_2	O_1	O_0
1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	0
×	0	1	1	1	1	1	1	1	1	1	0	1
×	×	0	1	1	1	1	1	1	1	1	0	0
×	×	×	0	1	1	1	1	1	1	0	1	1
×	×	×	×	0	1	1	1	1	1	0	1	0
×	×	×	×	×	0	1	1	1	1	0	0	1
×	×	×	×	×	×	0	1	1	1	0	0	0
×	×	×	×	×	×	×	0	1	0	1	1	1
×	×	×	×	×	×	×	×	0	0	1	1	0

Decoders

A decoder is a device that has two or more decimally numbered outputs. Importantly, only one output is ever asserted at a time. The decoder asserts the output with the number that is the same as the binary number applied to the inputs. In this way, decoders work in exactly the opposite way to encoders.

This might be better understood if we look at a block diagram of a decoder and consider its operation in a little more detail. Figure 8 below shows the block diagram of an *2-line-to-4-line* decoder.

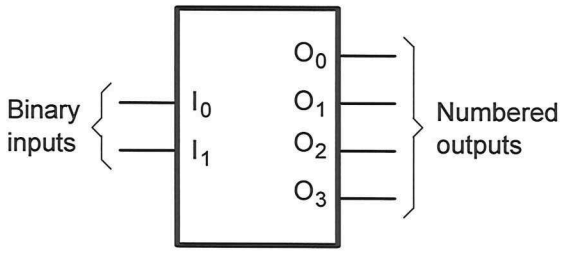


Figure 8 *The block diagram of a 2-line-to-4-line decoder*

As you can see, the device has four outputs numbered O_0, O_1, O_2 and O_3. The diagram indicates that this example of a decoder has outputs that are active high. This means that the logic level on the outputs is normally logic-0. When the device asserts an output it makes it logic-1. Decoders with active low outputs are also available.

The block diagram in Figure 8 also shows that the device has two inputs labelled I_0 and I_1 and together allow a 2-bit binary number (eg. 00, 01, 10 or 11) to entered into it.

The device's operation is as follows. When a binary number is applied to the input, the decoder asserts the output with the same number as the binary number. For example, if 00 is applied to the inputs then the device asserts Output-0, if 01 is applied to the inputs then the device asserts Output-1 and so on. This is shown in Figures 9a to 9d on the next page.

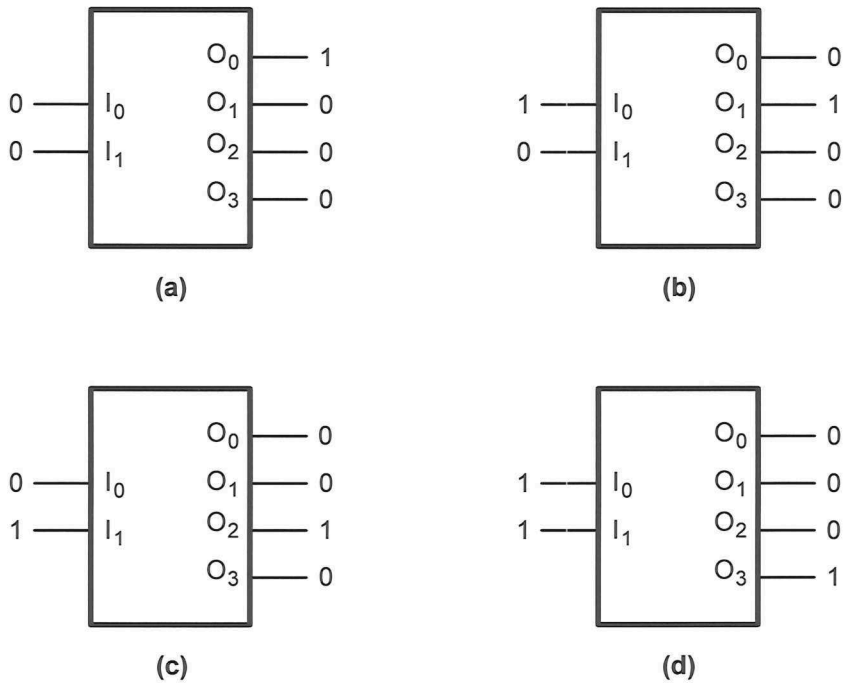


Figure 9 *a) When the binary number 00 is applied to the input, Output-0 is asserted; b) When the binary number 01 is applied to the input, Output-1 is asserted; c) When the binary number 10 is applied to the input, Output-2 is asserted; and d) When the binary number 11 is applied to the input, Output-3 is asserted*

This 2-line-to-4-line decoder is more commonly called a *1-of-4* decoder because it only ever asserts one of its four outputs. The logic circuit that implements this device is not shown here. However, the circuit and a brief explanation are provided in Appendix 2 for your interest.

The inputs to decoders are often called *select* inputs because the code on the inputs select which output is asserted.

Applications

An application for the decoder that can be shown here is a *light chaser*. A light chaser is a set of light bulbs or LEDs arranged in a line that are turned on one at a time either in a particular order or possibly just randomly. Light chasers find use in all sorts of situations and a simple example would be Christmas tree lights. The circuit in Figure 10 below shows how a string of four LEDs can be turned on, one at a time, one after the other. (Obviously, using decoders with more than just four outputs would be needed for a more impressive string of Christmas tree lights but you get the idea, hopefully.)

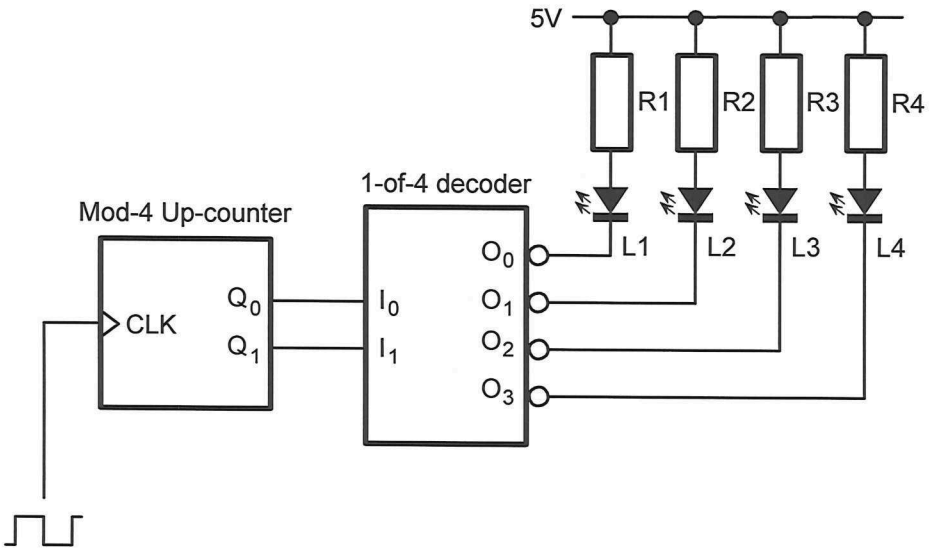


Figure 10 *The circuit of a four-LED light chaser*

The operation of this circuit is very straight forward. The clock signal causes the counter to continuously cycle through its count sequence from 00 to 11. As the counter's outputs are connected to the decoder's select inputs, this causes the decoder to continuously assert each output one at a time from O_0 to O_3. As asserting an output turns on an LED, the LEDs are continuously turned on one at a time from *L1* to *L4* over and over.

(Varying the clock signal's frequency will vary the speed of the LEDs along the turning on and off. And using an up-down counter that is made to count up to the highest number, but instead of overflowing, count back down to the lowest number, then back up and so on would produce would turn an LEDs that bounce from one end to the other then back again. And using a counter that counts up in even numbers would... I think you get the idea.)

A similar and more serious use for the light chaser is the floor indicator for elevators (like the one in Building L at the Ultimo campus of TAFE NSW). As the elevator moves up/down through the building's levels, an up/down-counter is clocked up turning on successive lamps to indicate which floor the elevator is on.

An important application for the decoder is a common display technique called *display multiplexing* that is used for multiple digit seven-segment displays. To explain the technique, consider the circuit in Figure 11 below.

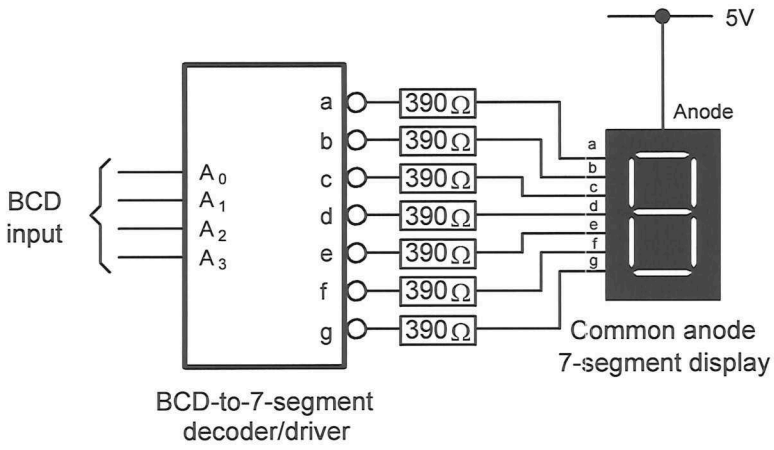


Figure 11 *A BCD-to-seven-segment decoder/driver driving a seven-segment display*

The BCD-to-seven-segment decoder/driver is used to control the seven-segment display (common anode in this example) and make it show the decimal equivalent of the BCD value on its inputs. It does so by decoding the BCD input and making its outputs logic-0 or logic-1 to turn the appropriate LEDs that make up the seven-segment display on or off.

So, for example, if 0000 is applied to the decoder/driver's input then the device makes output *g* a logic-1 and outputs *a*, *b*, *c*, *d*, *e* and *f* logic-0. Doing so forces the display to show the number 0. If 0001 is applied to the decoder/driver's input then the device makes outputs *a*, *d*, *e*, *f* and *g* logic-1 and outputs *b* and *c* logic-0 so the display shows the number 1 and so on.

Without display multiplexing, this drive circuitry must be duplicated for every digit in the readout. This is shown using the simple example of a two-digit readout in Figure 12 below.

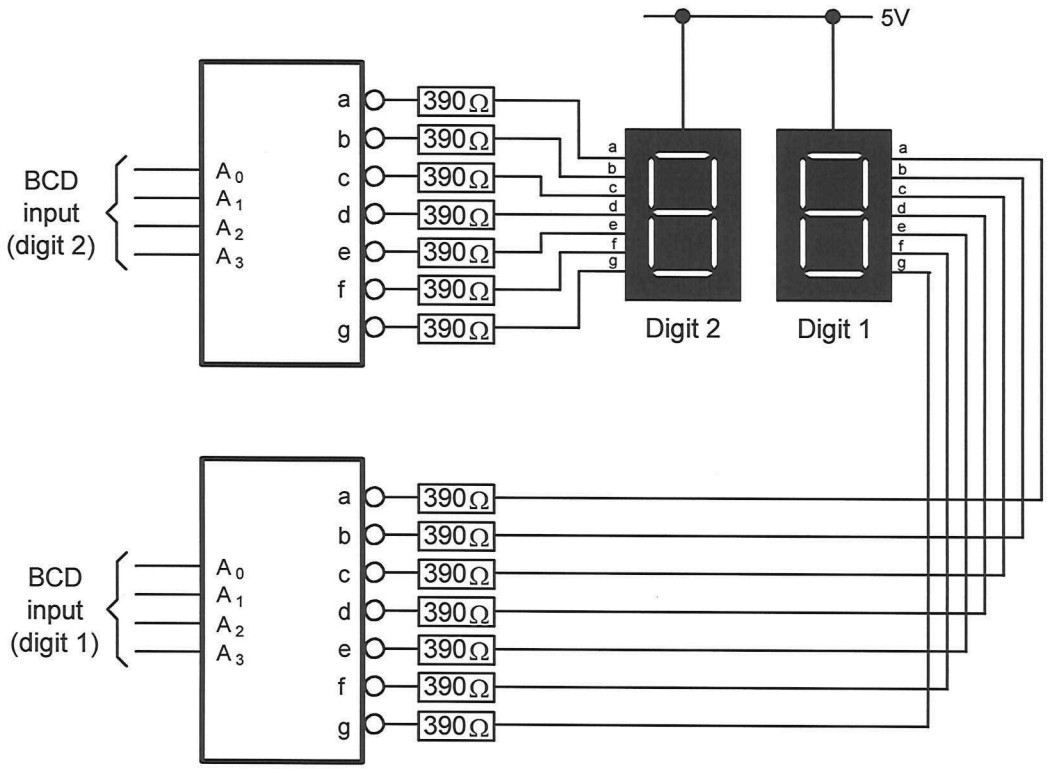


Figure 12 *A two digit readout using two seven segment displays*

As you can probably imagine, for every additional digit in the display the circuit becomes more and more busy and the printed circuit board (PCB) layout becomes increasingly complex making it difficult to design and manufacture.

With display multiplexing, there are two or more seven-segment displays in the readout but only one decoder/driver IC that is shared between all the displays. To manage the sharing, a decoder IC (like the 1-of-4 decoder) is used to control which display in the readout is connected to the decoder/driver. An example of a simple display multiplexing system is shown in Figure 13 on the next page.

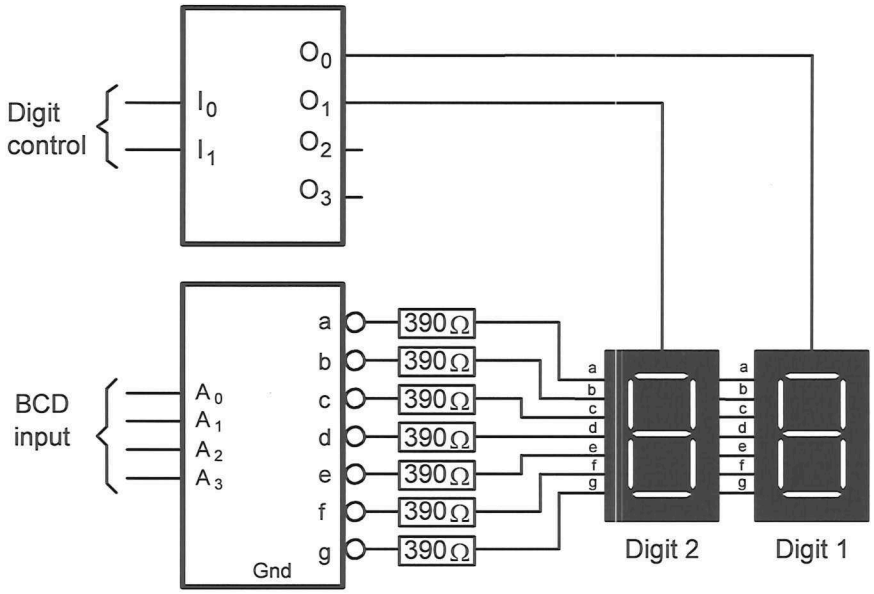


Figure 13 *Display multiplexing using a 1-of-4 decoder with active high outputs to control which digit is activated*

To explain how this circuit works, let's suppose that the display shows the number 39. First, the BCD code for the number nine is applied to the decoder/driver's input. This means that the device will output the appropriate logic levels to display the number nine. At the same time, the binary number 00 is applied to the select inputs of the 1-of-4 decoder. This makes O_0 a logic-1 enabling Digit-1 and so it displays the number nine. This is shown in Figure 14 below.

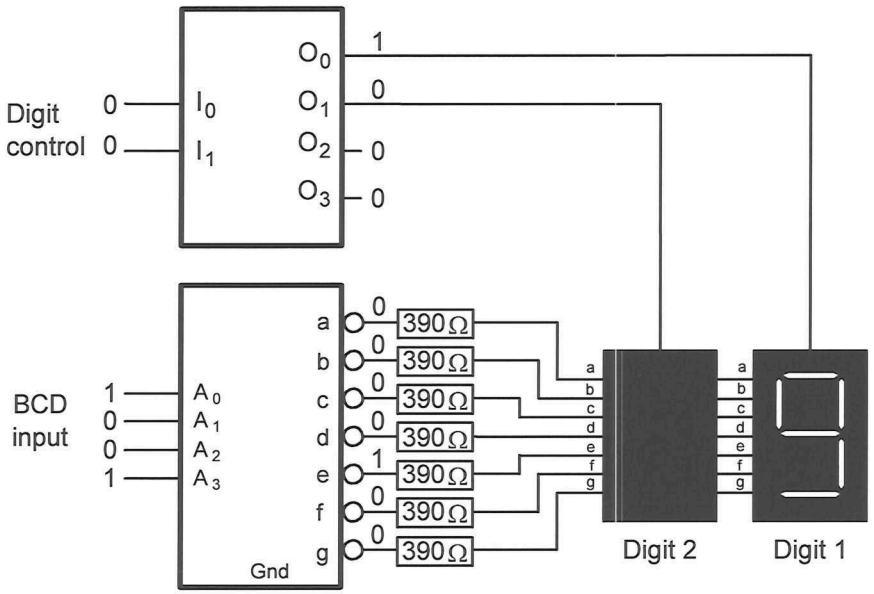


Figure 14

Then, the BCD code for the number three is applied to the decoder/driver's input. This means that the device will output the appropriate logic levels to display the number three. At the same time, the binary number 01 is applied to the select inputs of the 1-of-4 decoder. This makes O_1 a logic-1 enabling Digit-2 and so it displays the number three. This is shown in Figure 15 below.

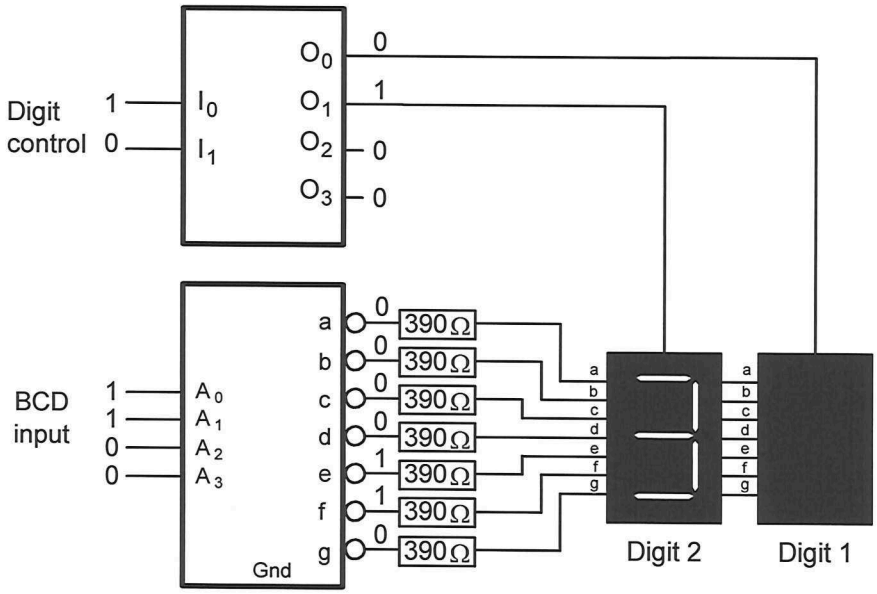


Figure 15

If the digits are switched fast enough they appear to be on all the time and so the display shows the number 39 instead of switching back-and-forth between 9 and 3.

Obviously a little more circuitry is needed than is shown in our simple example here to ensure that the decoder/driver gets the right BCD code at the right time. This means that display multiplexing requires more than just two chips and so it doesn't provide any real savings for a two-digit readout. However, for displays with four or more digits, display multiplexing uses fewer ICs and so is cheaper to make, cheaper to run, the display's PCB is smaller and simpler and therefore easier to design. Display multiplexing is usually used for the readouts of digital clock radios.

Decoder ICs

Decoding is a commonly used technique in digital electronics. Moreover, as you'll see, decoder ICs can also be used to implement another logic device. Consequently, there are a few commercially available decoder ICs. Examples include the 74138, 74139, 74154 and the 4028. To give you a feel for working with encoder ICs we'll look at just one of these devices and discuss its operation in a little detail.

The 74139 decoder

This TTL device is a dual 1-of-4 decoder (that is, there is two 1-of-4 decoders inside the IC) with active low outputs and an active low enable input. The pinout for the 74139 is shown in Figure 11 below. Notice that the labelling of the inputs and outputs is slightly different to what has been used so far. There are always variations in this regard between manufacturers' data.

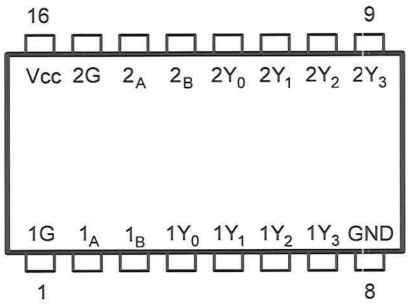


Figure 11 *Pinout data for the 74139*

To help make the device's operation a little easier to understand, it has been redrawn in Figure 12 below in block diagram form with the pins arranged more conveniently. As there are two decoders inside the IC there are two blocks in Figure 12.

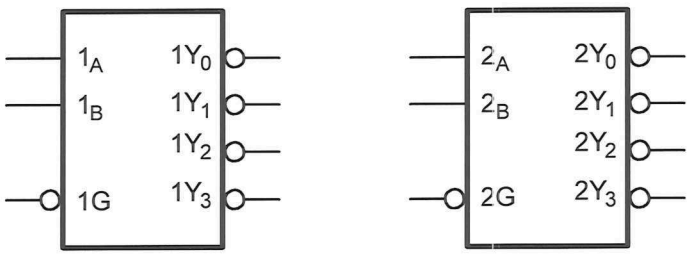


Figure 12 *Block diagrams for the decoders inside the 74139 with the pins visually reorganised*

Like the encoder IC that we've looked at earlier in this section, the wiring required to operate the 74139 is pretty straight forward and so a circuit is not shown. However, the decoders in the 74139 are a little different to the simple decoder in Figure 8 in two ways and these need to be explained.

First, as the block diagrams in Figure 12 show, the outputs are active low. That being the case, the outputs are usually logic-1 and the asserted output is a logic-0. This is shown in the device's truth table (Table 2 below).

Second, you'll notice that the devices have a gate input and, as the block diagrams show, this input is active low. Importantly, when the gate input is asserted, these 1-of-4 decoders work as they should. But when the gate inputs are not asserted, none of the outputs are asserted regardless of the logic levels on the select inputs (*A* and *B*). This is shown in the top row of Table 2 below.

Table 2

Inputs			Outputs			
G	B	A	Y_0	Y_1	Y_2	Y_3
1	×	×	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

Copyright: Not to be reproduced without permission

Student notes

Skill practice 6

Practise testing and measuring digital decoder ICs

This exercise is practise for the sorts of skills you may be required to perform in a practical test. Remember, in any practical tests you will be working alone so make sure that you can perform all the steps. It should take you about 1¼ hours to complete this exercise.

Equipment

- Digital trainer
- logic probe
- 74LS139
- 74LS193
- 74LS00
- hook-up wire (terminated)
- IC extractor

Remember:

Follow TAFE NSW WHS guidance at all times!

Work tasks

1. Read your WHS responsibilities at the top of the form below. Then conduct a WHS risk assessment and record your findings in the space provided.

Responsibilities of students under the Model WHS Act: s28

- Take reasonable care for your own health and safety by working safely at all times
- Take reasonable care to ensure that your acts or omissions don't put the health and safety of others at risk
- Follow all TAFE NSW WHS guidance and comply with all reasonable instructions from TAFE NSW staff to assist them in complying with the TAFE NSW WHS requirements
- In addition to the above, you must:
  - use and maintain machinery, tools and all other equipment properly and safely
  - ensure that your work area is free of hazards
  - notify a TAFE NSW staff member of actual or potential hazards
  - wear/use prescribed safety equipment
  - take notice of any safety signs and adhere to their instructions

Risks involved in this activity include:	**Control measures:**
■ Trip hazards (eg students bags) ■ Objects dropped on feet (while equipment is taken to and from benches).	■ Move bags and other objects from walkways ■ Plan lifting of equipment
Others: _____ _____ _____	Other: _____ _____ _____

My signature here indicates that I have read and understand my responsibilities under the Model WHS Act s28 (detailed above). I have also conducted a risk assessment before undertaking this activity and have identified measures to control these risks and have implemented them.

Signature: _____ Date: _____

2. Gather the equipment needed for this exercise.

3. Read the following points and make sure that you perform them when wiring all circuits:

   - Write the IC pin numbers of the devices you work with on the circuit that you're instructed to wire.
   - Handle all ICs carefully - remember ESD can damage them.
   - Ensure the power supply is connected to all ICs in the circuit.
   - When plugging hook-up wire into the breadboard to make connections to the IC's pins, use holes that are furthest away from the pin.

4. Wire the circuit of Figure 1 on the digital trainer ensuring that you connect the power supply to the IC. Pinout data for the 74LS139 can be found in Appendix 6.

   Tip: You'll be building on to this circuit so, to make that a little easier for you, insert the 74LS139 to the left-hand side of the breadboard.

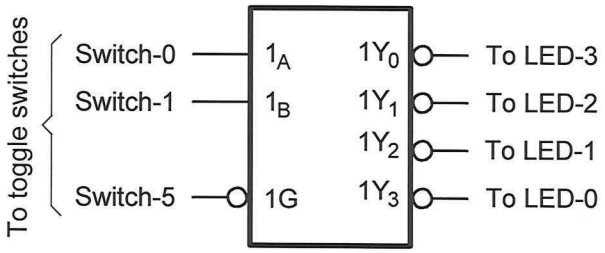


Figure 1

5. Turn on the trainer.

6. Use the toggle switches to make inputs *A* and *B* logic-0 and input *G* a logic-1.

7. Read the logic levels on the device's outputs and record them in the first row of Table 1 (on the next page).

8. Use the toggle switches to make the *A*, *B* and *G* inputs the same as the logic levels in the next row of Table 1 and repeat Step 6.

9. Repeat this procedure until the Table is complete.

Table 1

Inputs			Outputs			
G	B	A	Y_0	Y_1	Y_2	Y_3
1	0	0				
1	0	1				
1	1	0				
1	1	1				
0	0	0				
0	0	1				
0	1	0				
0	1	1				

Question 1
Does the number on the *A* and *B* inputs assert any of the device's outputs while *G* is a logic-1?

Question 2
Does the number on the *A* and *B* inputs assert any of the device's outputs while *G* is a logic-0?

Question 3
That being the case, which of the following statements is true?

☐ The G-input is active low

☐ The G-input is active high

Question 4
When an output is asserted, what logic level does it go to?

_____

Question 5
That being the case, which of the following statements is true?

☐ The outputs are active low

☐ The outputs are active high

Question 6
When the decoder is enabled, what is the relationship between the number on the *A* and *B* inputs and the asserted output?

_____

_____

_____

The teacher needs to check your work at this point...

10. Wire the circuit in Figure 2 below. Pinout data for the 74LS193 can be found in Appendix 6.

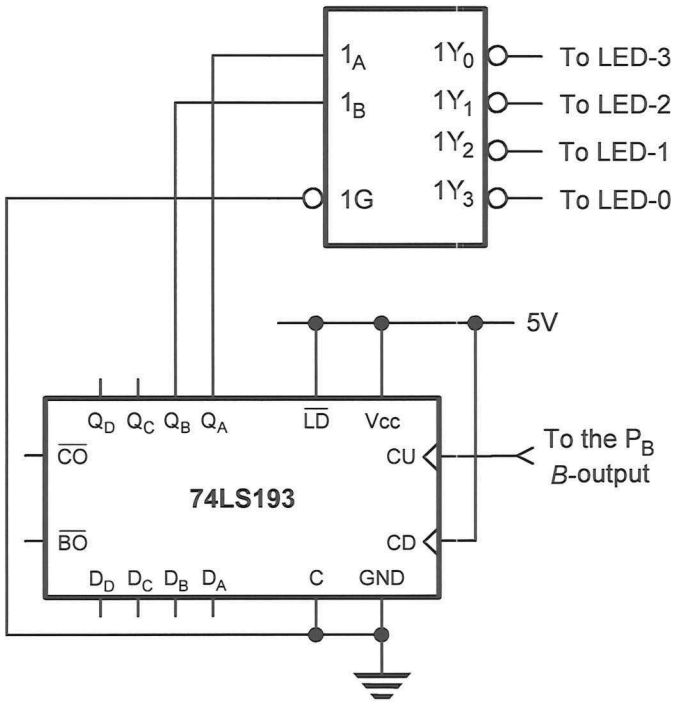


Figure 2

11. Press the pulser button a few times and observe what happens.

Self check: If your circuit is working correctly you should see the LEDs switch on and off one at a time in a particular order. If this doesn't happen, check your wiring. If you still can't get the circuit to work, call the teacher.

Question 7
What is the name of this circuit?

_____

Question 8
Briefly explain how you think the circuit works.

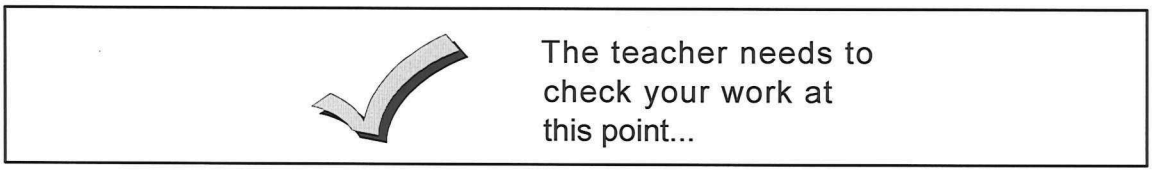
_____

_____

_____

Question 9

How can the circuit be modified so that the LEDs move from right to left (instead of left to right) without changing the physical order of the LEDs?



12. Wire the circuit in Figure 3 below. Pinout data for the 74LS00 can be found in Appendix 6.

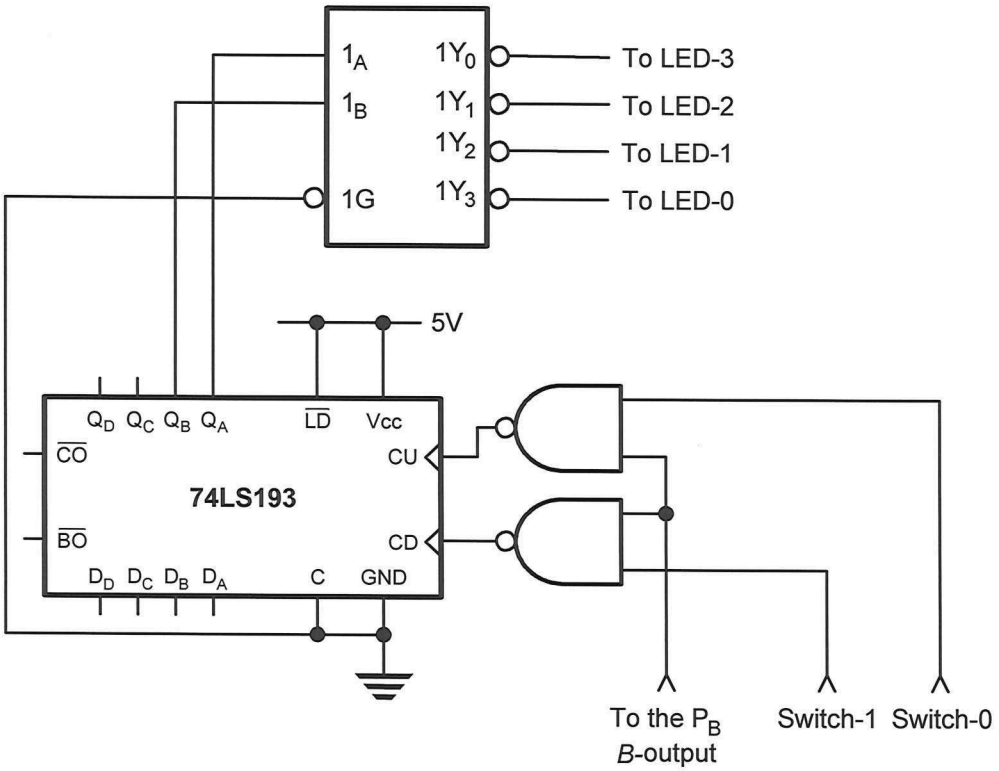


Figure 3

 Copyright: Not to be reproduced without permission

13. Set the toggle switches so that Switch-0 is logic-1 and Switch-1 is logic-0.

14. Press the pulser switch a few times and observe what happens.

    Self check: If your circuit is working correctly it should do the same as the circuit in Figure 2. If it doesn't, check your wiring. If you still can't get the circuit to work, call the teacher.

15. Set the toggle switches so that Switch-0 is logic-0 and Switch-1 is logic-1.

16. Press the pulser switch a few times and observe what happens.

    Self check: If your circuit is working correctly the LEDs should turn on and off one at a time but a little differently. If this doesn't happen, check your wiring. If you still can't get the circuit to work, call the teacher.

Question 10
Briefly explain how you think the switches and NAND gates together control whether the LEDs move from right to left or left to right.

The teacher needs to check your work at this point...

17. Wire the circuit in Figure 4 below.

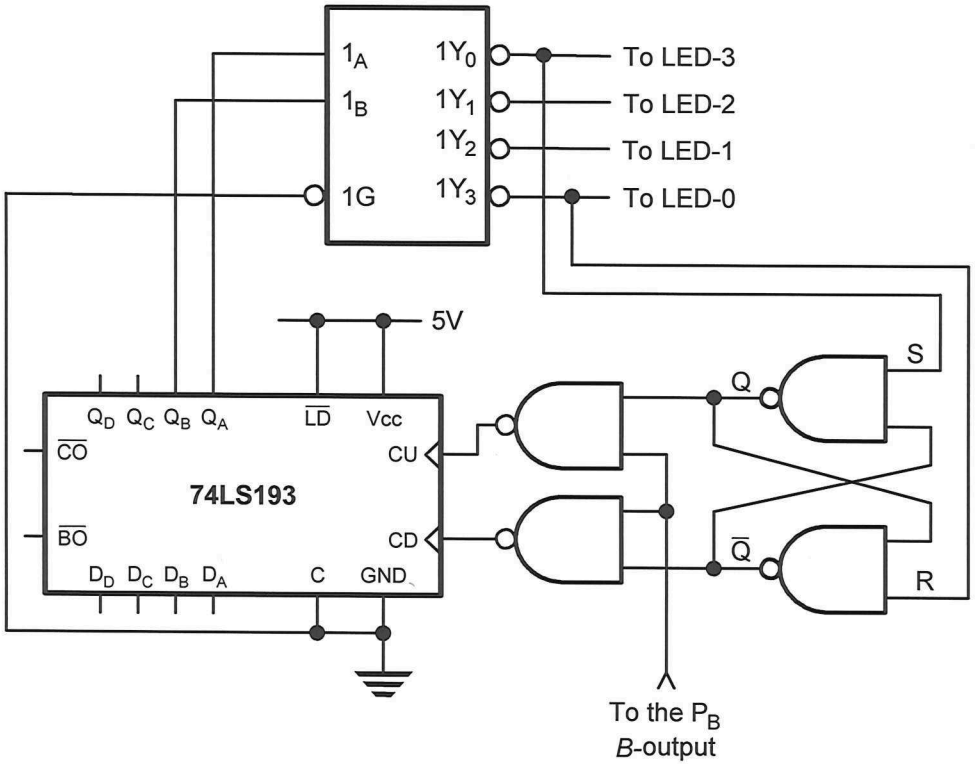


Figure 4

Self check: If your circuit is working correctly the LEDs should turn on and off one at a time moving left to right then right to left. If this doesn't happen, check your wiring. If you still can't get the circuit to work, call the teacher.

18. Move the wire from the trainer's P_B output to the *TTL MODE* clock source output and adjust the source's controls to appropriate settings so that you can watch the LEDs move left and right automatically.

Question 11
Briefly explain how you think the circuit works.

Copyright: Not to be reproduced without permission

The teacher needs to check your work at this point...

Review questions

Answer these questions to check your understanding of what you have learnt from this chapter. Doing this will also help to prepare you for the assessments.

1. What happens when one of the encoder's inputs is asserted?

2. What happens when a binary number is entered into a decoder?

3. Which device has more inputs than outputs?

   ☐ An encoder
   ☐ A decoder
   ☐ Both

4. Which device cannot have more than one output asserted at a time?

   ☐ An encoder
   ☐ A decoder
   ☐ Both

5. Which device can have inputs with that are prioritised?

   ☐ An encoder
   ☐ A decoder
   ☐ Both

6. Which device can have either active low or active high outputs?

   ☐ An encoder
   ☐ A decoder
   ☐ Both

Copyright: Not to be reproduced without permission

7. List two applications for encoders.

_____

_____

8. List three applications for decoders.

_____

_____

_____

Questions 9 to 16 refer to the device shown in Figure 1 and its truth table

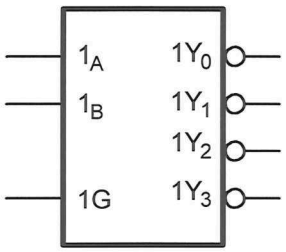


Figure 1

Inputs			Outputs			
G	B	A	Y_0	Y_1	Y_2	Y_3
0	×	×	1	1	1	1
1	0	0	0	1	1	1
1	0	1	1	0	1	1
1	1	0	1	1	0	1
1	1	1	1	1	1	0

9. The device is

- ☐ an encoder.
- ☐ a priority encoder.
- ☐ a decoder.
- ☐ a priority decoder.

10. The gate is

- ☐ active low.
- ☐ active high.
- ☐ positive edge-triggered.
- ☐ negative edge-triggered.

11. The other two inputs are

- ☐ active low.
- ☐ active high.
- ☐ positive edge-triggered.
- ☐ negative edge-triggered.

12. The outputs are

- ☐ active low.
- ☐ active high.
- ☐ positive edge-triggered.
- ☐ negative edge-triggered.

13. Why are all of the outputs logic-1 in the first row of the truth table?

14. Why is there an × in the *A* and *B* columns of the same row?

15. What is the relationship between the number on the *A* and *B* inputs and the number of the asserted output when the device is enabled?

16. Which of the following statements about the device is true?

- ☐ No more than one input can be asserted at a time
- ☐ No more than one output is asserted at a time
- ☐ If more than one input is asserted, the device only accepts the highest weighted input
- ☐ The logic levels on the outputs form a binary number

 Copyright: Not to be reproduced without permission

Questions 17 to 23 refer to the device shown in Figure 2 and its truth table

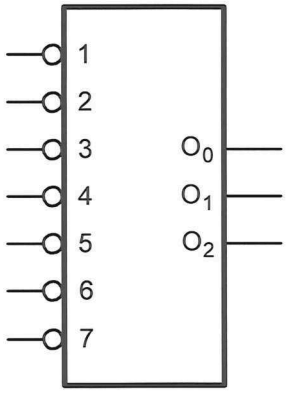


Figure 2

Inputs							Outputs		
1	2	3	4	5	6	7	O_2	O_1	O_0
1	1	1	1	1	1	1	0	0	0
0	1	1	1	1	1	1	0	0	1
×	0	1	1	1	1	1	0	1	0
×	×	0	1	1	1	1	0	1	1
×	×	×	0	1	1	1	1	0	0
×	×	×	×	0	1	1	1	0	1
×	×	×	×	×	0	1	1	1	0
×	×	×	×	×	×	0	1	1	1

17. The device is

- ☐ an encoder.
- ☐ a priority encoder.
- ☐ a decoder.
- ☐ a priority decoder.

18. The inputs are

- ☐ active low.
- ☐ active high.
- ☐ positive edge-triggered.
- ☐ negative edge-triggered.

19. The outputs are

- ☐ active low.
- ☐ active high.
- ☐ positive edge-triggered.
- ☐ negative edge-triggered.

20. Why doesn't the logic level on input-1 matter whenever any of the other inputs are asserted?

21. What is the relationship between the asserted input and the number that appears on the outputs?

22. What happens when none of the inputs are asserted?

23. Which of the following statements about the device is true?

- ☐ More than one input can be asserted at a time
- ☐ If more than one input is asserted, all but one input is ignored
- ☐ The logic levels on the outputs form a binary number
- ☐ All of the above

24. How many outputs does a 16-line-to-4 line encoder have?

- ☐ 2
- ☐ 3
- ☐ 4
- ☐ 16

25. How many inputs does a 16-line-to-4 line encoder have?

- ☐ 2
- ☐ 3
- ☐ 4
- ☐ 16

 Copyright: Not to be reproduced without permission

26. How many inputs does a 1-of-8 decoder have?

- ☐ 1
- ☐ 3
- ☐ 4
- ☐ 8

27. How many outputs does a 1-of-8 decoder have?

- ☐ 1
- ☐ 3
- ☐ 4
- ☐ 8

Questions 28 to 31 refer to the 74LS138 IC. Data for the device can be found in Appendix 6

28. The device inside the IC is

- ☐ a 1-of-8 decoder with active low outputs.
- ☐ an 8-line-to-3-line decoder with active high inputs.
- ☐ a 1-of-8 decoder with active high outputs.
- ☐ an 8-line-to-3-line decoder with active low inputs.

29. How many enable inputs does the device have?

- ☐ 0
- ☐ 1
- ☐ 2
- ☐ 3

30. What logic level must the enable inputs be to enable the device?

- ☐ All three enable inputs must be logic-0
- ☐ *G1* must be logic-0 and *G2A* & *G2B* must be logic-1
- ☐ *G1* must be logic-1 and *G2A* & *G2B* must be logic-0
- ☐ All three enable inputs must be logic-1

31. What is the device's typical propagation delay?

_____

Questions 32 to 36 refer to the 74LS148 IC. Data for the device can be found on the internet (ty the Element 14 website)

32. The device inside the IC is

- ☐ a 1-of-8 encoder with active low outputs.
- ☐ an 8-line-to-3-line encoder with active high inputs.
- ☐ a 1-of-8 encoder with active high outputs.
- ☐ an 8-line-to-3-line encoder with active low inputs.

33. How many enable inputs does the device have?

- ☐ 0
- ☐ 1
- ☐ 2
- ☐ 3

34. The enable input is

- ☐ active low.
- ☐ active high.
- ☐ negative edge-triggered.
- ☐ positive edge-triggered.

 Copyright: Not to be reproduced without permission

35. Apart from the enable input and the power supply connections, how many other inputs does the device have?

- ☐ 3
- ☐ 5
- ☐ 7
- ☐ 8

36. The outputs are

- ☐ active low.
- ☐ active high.
- ☐ negative edge-triggered.
- ☐ positive edge-triggered.

Section 7 Multiplexers and demultiplexers

Purpose To develop your understanding of multiplexers and demultiplexers and to develop your ability to interpret the manufacturers' data sheets for common commercially available multiplexer and demultiplexer ICs.

Objectives At the end of this section you should be able to:

- Define the term *time-division multiplexing*
- Explain the basic operation of a multiplexer and demultiplexer (at the block diagram level only)
- Interpret the truth table of multiplexers and demultiplexers
- Give typical applications for multiplexers and demultiplexers
- Interpret manufacturers' data for common commercially available multiplexer and demultiplexer ICs
- Use a logic probe to test and measure multiplexers and demultiplexers

Introduction

This chapter looks at the operation of another set of digital devices called *multiplexers* and *demultiplexers*. Their general principles of operation are discussed as well as typical applications. And, the operation of a common commercially available multiplexer and demultiplexer ICs is explained. However, first you must learn about the general concept of multiplexing.

Multiplexing

The word *multiplex* is simply a technical way of saying "sharing". It is derived from the word "multiple" because that's what sharing involves: multiple users accessing the one resource.

In electronics the shared resource is usually a transmission medium like the "airwaves" (actually *free space*) or a transmission cable. There are several ways that these media can be multiplexed, two are:

- frequency-division multiplexing
- time-division multiplexing

Frequency-division multiplexing involves two or more users sharing the airwaves. This is done by giving each user their own radio frequency to use as a *carrier* (for example, Triple-J's carrier is 105.7MHz). Users superimpose their signal (speech, music and/or video) onto their carrier then transmit it. This allows multiple users to operate in the same geographic area without interfering with each other. This method of sharing is used for all free-to-air analog radio and television broadcasts as well as most other analog RF communications like CB radio, shortwave radio and emergency services communications. We'll not discuss this method of multiplexing any further in this unit (you'll learn much more about this in another unit).

Time-division multiplexing involves two or more users sharing a transmission medium (such as a cable) by taking turns at having exclusive use of the medium for short periods of time. This is a little like a couple who own one car between them. While one is driving the car, the other cannot.

As an aside, communication systems that use both frequency and time division multiplexing are possible. In fact, such systems are used extensively in telecommunications including the data transfer between digital mobile phones and their base stations (cells).

A simple mechanical time-division multiplexed system

The basic idea of time-division multiplexing can be demonstrated using a mechanical system for transmitting multiple telephone conversations down a single line. In such a system, pairs of users are connected to each other via two sets of continuously rotating switches. An example of this with 4-position rotary switches is shown in Figure 1 below.

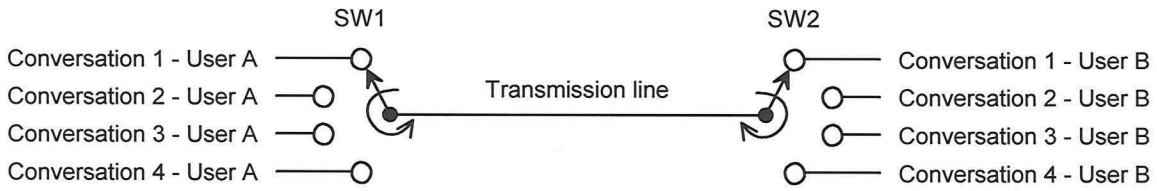

SW1
SW2
Conversation 1 - User A
Conversation 2 - User A
Conversation 3 - User A
Conversation 4 - User A
Transmission line
Conversation 1 - User B
Conversation 2 - User B
Conversation 3 - User B
Conversation 4 - User B

Figure 1 *A mechanical time-division multiplexed telephone system*

As Figure 1 shows, users *A* and *B* in conversation-1 are connected to each other. But, as the switches are rotating, this connection is only momentary. In fact, the users in all the conversations are only connected to each other for very brief amounts of time which means that only small *samples* of each conversation are sent down the transmission line! However, if the switches rotate fast enough (there is a fairly simple equation for working out how fast they should rotate) all the conversations will sound perfectly normal to the users.

Importantly, in this arrangement, switch *SW1* is a mechanical multiplexer. It connects only one of several inputs to a shared output. Switch *SW2* is a mechanical demultiplexer. It connects a shared input to only one of several outputs.

As an aside, it is clear from this simple mechanical time-division multiplexing system than the action of the two switches must be correctly synchronised. User-A in conversation-1 must be connected to user-B in conversation-1. Moreover, when the two switches change this must occur at the same time otherwise both users will hear parts of other conversations (which is one cause of a *crossed-line* in telecommunications). Although it's not discussed in this unit, synchronisation in digital time-division multiplexed systems is just as important as in the mechanical system above. You'll learn more about synchronisation in one of the electives.

Multiplexers

In digital electronics, a multiplexer (often just called a *mux*) is a device that takes digital information (either binary numbers or codes like ASCII) from just one of several inputs and copies it to a shared output.

The block diagram of a simple 2-input multiplexer is shown in Figure 2 below.

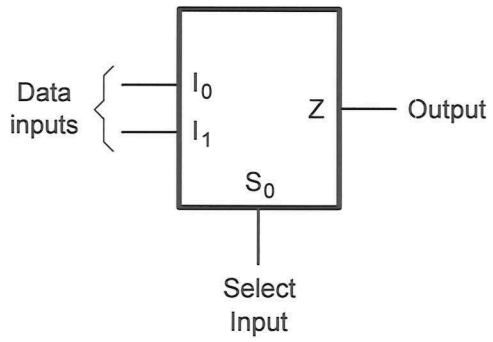


Figure 2 *A 2-input mux*

As you can see, the device has two inputs but only one of them gets copied to the output at a time. Control over this is provided by the *select* input (S_0).

The operation of the select input is straight forward. The data on the input with the decimal number equivalent to the binary number on the select input gets copied to the output. So, if S_0 is a logic-0 then the data on input I_0 is copied to the output. If S_0 is a logic-1 then the data on input I_1 is copied to the output. This is shown in Figure 3 below.

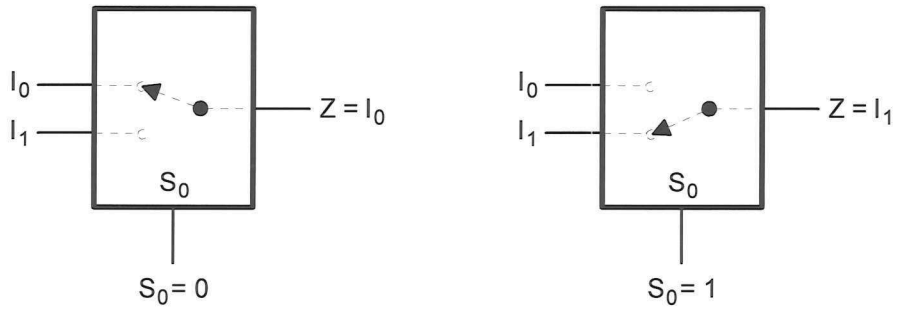


Figure 3 *The relationship between the logic level on the select input and the input that gets copied to the output*

(Figure 3 shows switches being used but the actual implementation of the mux is done using a simple combinational logic circuit not shown here. However, the circuit and a brief explanation of it are provided in Appendix 3 for your interest.)

The operation of the device can also be described using a truth table as shown in Table 1 below.

Table 1

Inputs			Output
S_0	I_0	I_1	Z
0	0	×	0
0	1	×	1
1	×	0	0
1	×	1	1

This truth table is often simplified as shown in Table 2 below.

Table 2

S_0	Z
0	I_0
1	I_1

An important point to make about the example of a multiplexer in Figure 2 is that the information on the inputs and output must necessarily be serial data because one bit only ever goes in and out of the multiplexer at a time.

There are multiplexers that handle parallel data. Such multiplexers are said to have two or more input *channels* only one of which gets copied to a single output channel. The number of bits in each input and the output are the same. For example, if the input channels are 4-bits wide then the output channel must also be 4-bits wide. An example of a 4-bit, 2-channel mux is shown in Figure 4 on the next page.

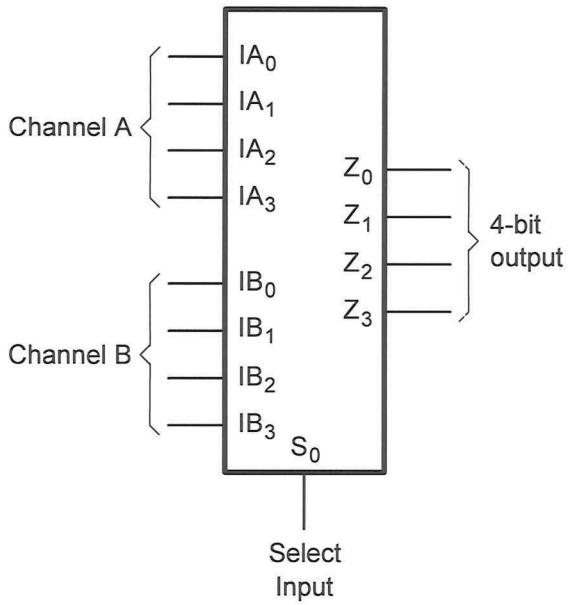


Figure 4 *A 2-channel mux*

Control over which channel is copied to the output is still managed by a select input. When the select input (S_0) is a logic-0 the data on each bit of channel-A is copied to the corresponding output bit. That is, IA_0 is copied to Z_0, IA_1 is copied to Z_1 and so on. When the select input is a logic-1 the data on each bit of channel-B is copied to the corresponding output bit.

Applications
Three commonly used applications for the multiplexer include:

- data routing
- parallel-to-serial conversion
- display multiplexing

Let's consider each of these in turn.

Data routing
Data routing is a hardware method of digitally directing data from one of several sources of data to one or more destinations. The simplest example of this is a *data switch-box* that can be used to allow two or more computers to share a printer (before the advent of computer networking).

To use a switch-box, the printer-output of each computer is connected to inputs on the switch-box. The switch-box has only one output that feeds the input to the printer. This is shown in Figure 5 below.

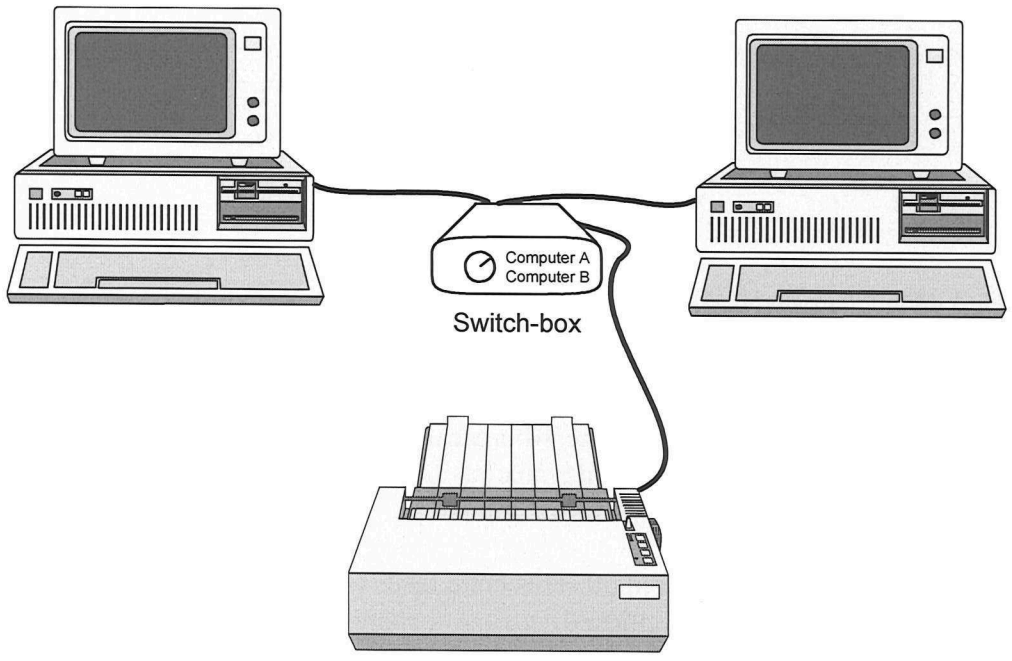


Figure 5 *Two computers sharing a printer via a data switch-box*

Inside the switch-box is a multiplexer IC and it is used to electrically route data to the printer from either computer-A or computer-B. The control on the switch-box's front-panel is simply a switch that changes the logic level on the select input of the mux chip.

Hardware data routing for this particular application has lost favour since the advent of local area networks (LANs). These use software data routing techniques that have much greater flexibility. However, hardware multiplexing is still used in other applications that are not as easy to explain as this example.

Parallel-to-serial conversion

As you know, a parallel in, serial out shift register can be used to convert data from parallel form to serial form. However, with the help of a counter, a multiplexer can also be used to do the job. Figure 6 below shows the circuit that converts 4-bit parallel data words into serial form.

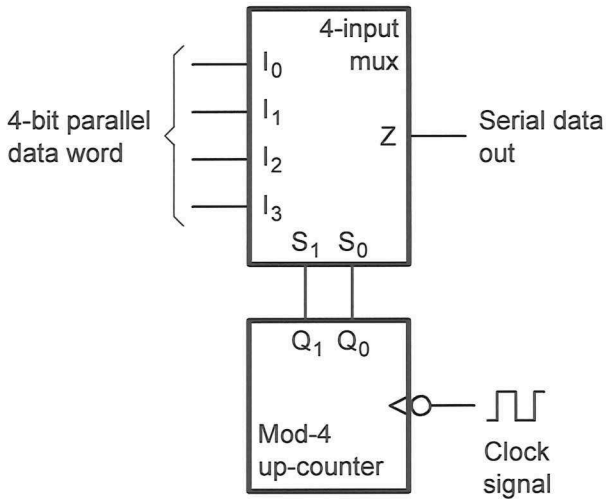


Figure 6 *A 4-input mux and a Mod-4 up-counter combine to make a circuit that converts 4-bit parallel data words into serial form*

The circuit's operation is as follows. The 4-bit parallel data word to be converted is applied to the inputs of the 4-input mux. [Note: As this mux has four data inputs, it must also have two select inputs - S_0 and S_1 - to provide the necessary control.]

Let's assume the counter starts from 00. This code feeds the select inputs of the mux and I_0 is copied to Z. When the counter is clocked, its output changes to 01 and this forces the mux to copy the I_1 to Z. On the next clock pulse the counter's output changes to 10 and this code forces the mux to copy the I_2 to Z. Then on the next clock pulse, the counter's output changes to 11 and this code forces the mux to copy the I_3 to Z.

In this way, the bits that make up the parallel data on the inputs to the mux appear one bit at a time on its output and so it is converted to serial form.

Another clock pulse causes the counter to overflow forcing the mux to copy I_0 to Z again. If a new parallel data word is to be converted, the data must change just before the counter overflows.

Display multiplexing

The previous chapter introduced you to the basic principle of display multiplexing and how a decoder can be used to help implement it. A copy of that circuit is shown in Figure 7 below.

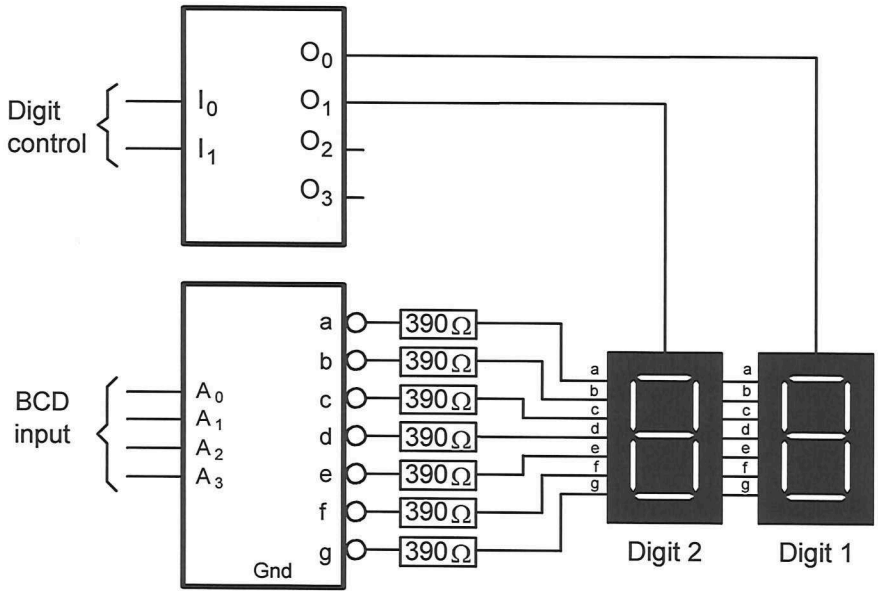


Figure 7 *The example of display multiplexing given in Section 7*

But when you think about it, Figure 7 must be incomplete. That's because the information to be displayed by the two digits comes from two separate sources (for example, the number on Digit-1 could comes from a counter counting units and the number on Digit-2 comes from a different counter counting tens of units). The 7-segment decoder/driver has only one input so the data from these two sources must be delivered to it via a multiplexer. This is shown in Figure 8 on the next page.

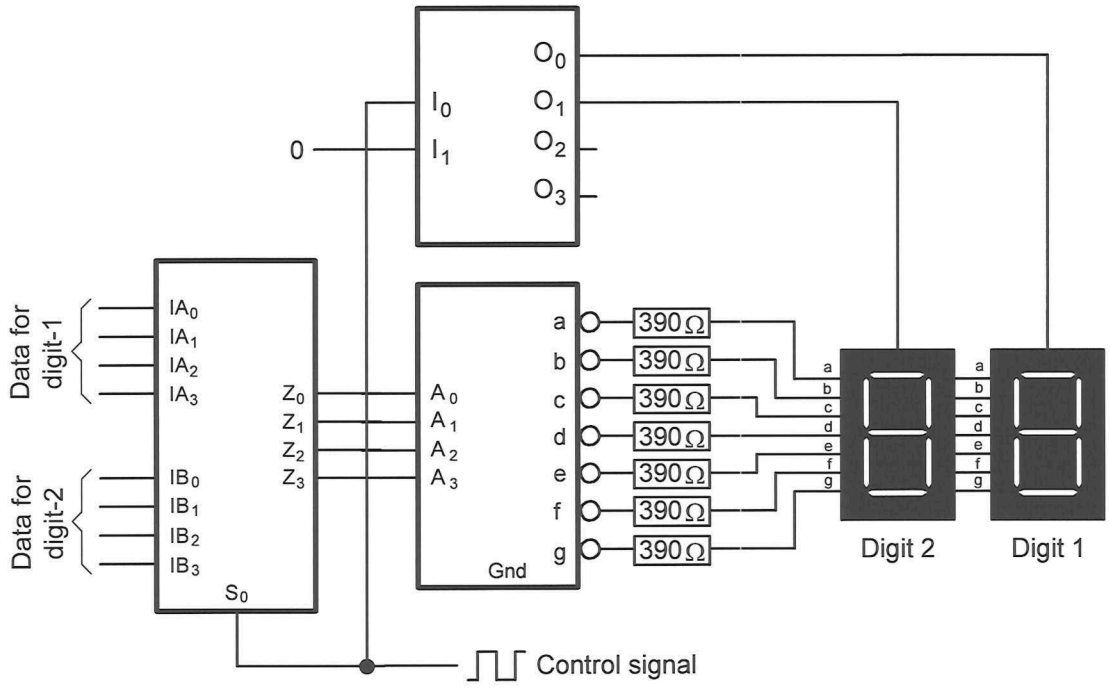


Figure 8 *The complete circuit for a 2-digit display multiplexed system*

When the control signal is a logic-0, the select input of the mux (S_0) is logic-0 and the inputs to the 1-of-4 decoder (I_0 and I_1) are 00. This means that data on the multiplexer's channel-A input is copied to its output and this information becomes the data input to the 7-segment decoder/driver. At the same time, O_0 is asserted and so Digit-1 is enabled and displays that number.

When the control signal is a logic-1, the select input of the mux (S_0) is logic-1 and the inputs to the 1-of-4 decoder (I_0 and I_1) are 01. This means that data on the multiplexer's channel-B input is copied to its output and this information becomes the data input to the 7-segment decoder/driver. At the same time, O_1 is asserted and so Digit-2 is enabled and displays that number.

Clearly, a full display multiplexing system requires quite a few ICs and so there are only savings in this regard when it's used for displays with four or more digits. Having said that, display multiplexing can still be used for displays with less than four digits to reduce the complexity of the circuit board.

Multiplexer ICs

There are many multiplexer ICs and examples include the 74151, 74153, 74157, 74158, 74251 and the 74253. To give you a feel for working with multiplexer ICs we'll look at just one of these devices and discuss its operation in a little detail.

The 74151 multiplexer

This TTL device is an eight input, active low enabled multiplexer with complementary outputs. The pinout for the 74151 is shown in Figure 9 below. Notice that the labelling of the inputs and outputs is different to what has been used so far. There are always variations in this regard between manufacturers' data.

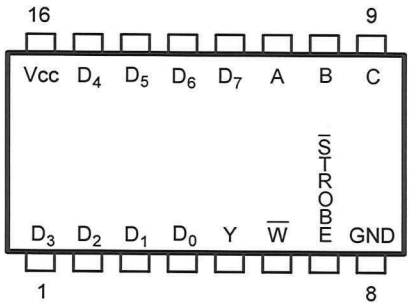


Figure 9 *Pinout data for the 74151*

To help make the device's operation a little easier to understand, it has been redrawn in Figure 10 below in block diagram form with the pins arranged more conveniently.

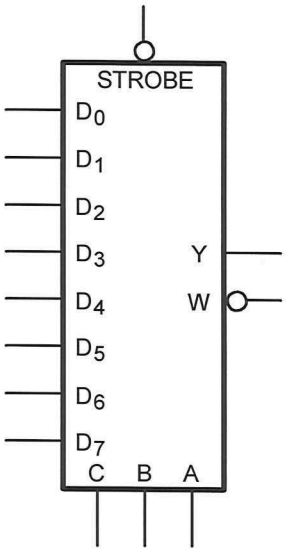


Figure 10 *A block diagram for the 8-input mux inside the 74151*

This device has only one mode of operation and so wiring the IC is pretty straight forward and is not shown here. However, the multiplexer inside the 74151 is a little different to the simple multiplexer in Figure 2 in three ways and these need to be explained.

First, the device has three select inputs (*A*, *B* and *C*) instead of just one. The reason for this relates to the fact the device has eight data inputs D_0 to D_7. To be able to select one of eight inputs a three-bit select code is required. The number 000 selects input D_0, 001 selects D_1 and so on up to 111 which selects D_7. This is shown in the device's truth table in Table 3 below.

Second, the device has *complementary* outputs (*Y* and *W*). That is, there are two outputs with *Y* always being the same logic level as the selected input (when the device is enabled) while *W* is the opposite of the logic level on the selected input. This is a little like the *Q* and $\overline{Q}$ outputs of a flip flop and is shown in Table 3 also.

Third, the multiplexer inside the 74151 has an enable input called *STROBE*. The strobe input is active low and so the device works as a mux when a logic-0 is applied to it. When a logic-1 is applied, the device ignores the select inputs and the *Y* output goes to logic-0 while the *W* output goes to logic-1. This is shown in Table 3 also.

Table 3

Inputs				Outputs	
Strobe	Select				
	C	B	A	Y	W
1	×	×	×	0	1
0	0	0	0	D_0	$\overline{D}_0$
0	0	0	1	D_1	$\overline{D}_1$
0	0	1	0	D_2	$\overline{D}_2$
0	0	1	1	D_3	$\overline{D}_3$
0	1	0	0	D_4	$\overline{D}_4$
0	1	0	1	D_5	$\overline{D}_5$
0	1	1	0	D_6	$\overline{D}_6$
0	1	1	1	D_7	$\overline{D}_7$

Demultiplexers

In digital electronics, a demultiplexer (often just called a *demux*) is a device that takes digital information (either binary numbers or codes like ASCII) from one shared input and copies it to one of several outputs.

The block diagram of a simple 2-output demultiplexer is shown in Figure 11 below. The logic circuit that implements this device is not shown here. However, the circuit and a brief explanation are provided in Appendix 3 for your interest.

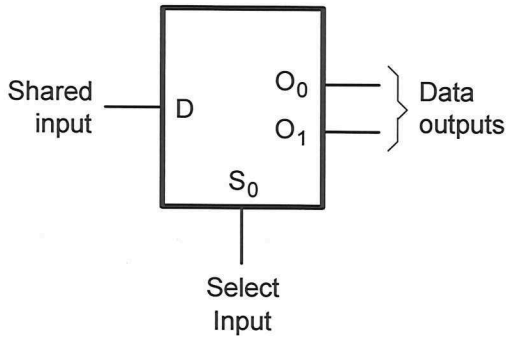


Figure 11 *A 2-output demux*

As you can see, the device has two outputs but only one of them is a copy of the input. Control over this is provided by the *select* input (S_0).

The operation of the select input is straight forward. The data on the input is copied to the output with the decimal number equivalent to the binary number on the select input. So, if S_0 is a logic-0 then the input data is copied to output O_0. If S_0 is a logic-1 then the input data is copied to output O_1. As the outputs are active high, the unselected output is always logic-0. This is shown in Figure 12 below.

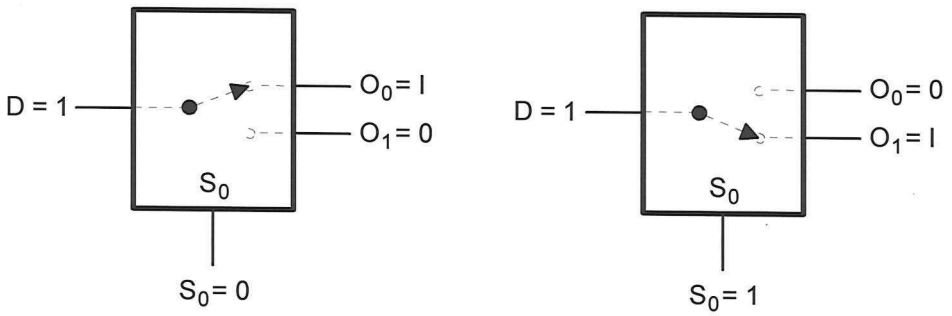


Figure 12 *The relationship between the logic level on the select input and the output that the input data gets copied to*

The operation of the device can also be described using a truth table as shown in Table 4 below.

Table 4

Inputs		Output	
S_0	D	O_0	O_1
0	0	0	0
0	1	1	0
1	0	0	0
1	1	0	1

This truth table is often simplified as shown in Table 5 below.

Table 5

S_0	O_0	O_1
0	D	0
1	0	D

An important point to make about the example of a demultiplexer in Figure 11 is that the data on the input and outputs are serial data because one bit only ever goes in and out of the demultiplexer at a time.

There are demultiplexers that handle parallel data. Such devices are said to have a single input channel that is copied to just one of two or more output channels. The number of bits in the input channel and the output channels are the same. For example, if the input is 4-bits wide then the output channels must also be 4-bits wide.

Applications
Two applications for the demultiplexer include:

- data routing
- serial-to-parallel conversion

The general concept of data routing was explained earlier. A demultiplexer is used when data from a single source needs to be distributed to several destinations.

As you know, a serial in, parallel out shift register can be used to convert data from serial form to parallel form. However, with the help of a counter and a D-type flip flop for every output, a demultiplexer can also be used to do the job. The circuit isn't shown here as this method generally requires more chips to implement than other methods of serial-to-parallel conversion.

Demultiplexer ICs

Interestingly, as you are about to see, decoder ICs also work as demultiplexers and don't require any special wiring or external circuitry to do so. Consequently, there are few (if any) dedicated demux ICs. Decoder ICs are usually described by their manufacturers as decoder/demultiplexers and examples of this include the 74138, 74139 and the 74154. All of these ICs are mentioned in the notes on decoders in Section 4 and the 74139 is discussed in detail.

To explain how decoder ICs are used for demultiplexing, an example of a 1-of-4 decoder operated as 4-output demux will be shown. As you know a little about the devices inside the 74139, we'll use that one. The block diagram of this device is shown in Figure 13 below and its truth table is shown in Table 6 beside it.

Figure 13

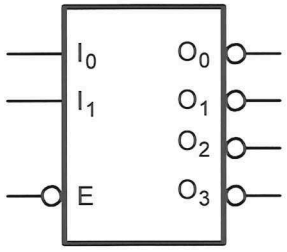


Table 6

Inputs			Outputs			
E	I_1	I_0	O_0	O_1	O_2	O_3
1	×	×	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

To use this decoder as a demultiplexer, simply treat the select inputs (I_0 and I_1) as though they are select inputs to a demux and treat the enable input as though it's the data input to a demux. This is shown in Figure 14 below. That's it! That's all there is to operating a 1-of-4 decoder as a 4-output demux.

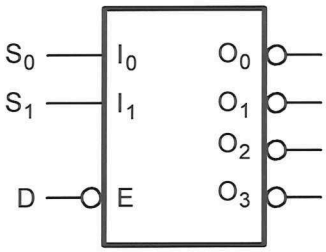


Figure 14 *A 1-of-4 decoder operated as a 4-output demux*

To demonstrate that this circuit operates as a demux, let's examine its operation in a little detail. Figure 15a below shows the logic levels on the outputs (according to the device's truth table) when 00 is applied to the select inputs and a logic-0 on the D-input. Figure 15b shows the logic levels on the outputs when 00 is applied to the select inputs and a logic-1 on the D-input.

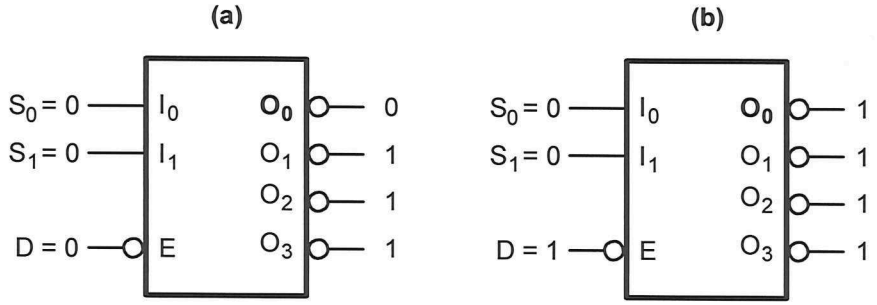


Figure 15 *a) With 00 on the select inputs and logic-0 on the D-input O_0 is logic-0; and b) With 00 on the select inputs and logic-1 on the D-input O_0 is logic-1*

As you can see from this example, when the select inputs are 00, O_0 is the same as the D-input. Let's see what happens with a different number on the select happens.

Figure 16a below shows the logic levels on the outputs when 10 is applied to the select inputs and a logic-0 on the D-input. Figure 16b shows the logic levels on the outputs when 10 is applied to the select inputs and a logic-1 on the D-input.

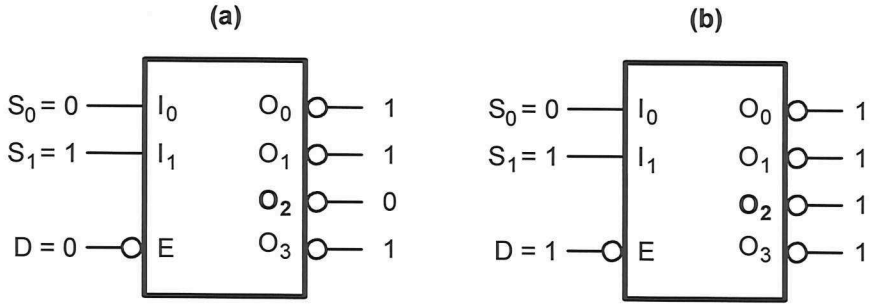


Figure 16 *a) With 10 on the select inputs and logic-0 on the D-input O_2 is logic-0; and b) With 10 on the select inputs and logic-1 on the D-input O_2 is logic-1*

As you can see from this example, when the select inputs are 10, O_2 is the same as the D-input. Clearly, the decoder is operating as a demultiplexer.

Copyright: Not to be reproduced without permission

Student notes

Skill practice 7

Practise testing and measuring digital multiplexer and demultiplexer ICs

This exercise is practise for the sorts of skills you may be required to perform in a practical test. Remember, in any practical tests you will be working alone so make sure that you can perform all the steps. It should take you about 1¼ hours to complete this exercise.

Equipment

- Digital trainer
- logic probe
- 74LS151
- 74LS139
- hook-up wire (terminated)
- IC extractor

Remember:

Follow TAFE NSW WHS guidance at all times!

 Copyright: Not to be reproduced without permission

Work tasks

1. Read your WHS responsibilities at the top of the form below. Then conduct a WHS risk assessment and record your findings in the space provided.

Responsibilities of students under the Model WHS Act: s28

- Take reasonable care for your own health and safety by working safely at all times
- Take reasonable care to ensure that your acts or omissions don't put the health and safety of others at risk
- Follow all TAFE NSW WHS guidance and comply with all reasonable instructions from TAFE NSW staff to assist them in complying with the TAFE NSW WHS requirements
- In addition to the above, you must:
  - use and maintain machinery, tools and all other equipment properly and safely
  - ensure that your work area is free of hazards
  - notify a TAFE NSW staff member of actual or potential hazards
  - wear/use prescribed safety equipment
  - take notice of any safety signs and adhere to their instructions

Risks involved in this activity include:	**Control measures:**
■ Trip hazards (eg students bags) ■ Objects dropped on feet (while equipment is taken to and from benches).	■ Move bags and other objects from walkways ■ Plan lifting of equipment
Others: _____ _____ _____	Other: _____ _____ _____

My signature here indicates that I have read and understand my responsibilities under the Model WHS Act s28 (detailed above). I have also conducted a risk assessment before undertaking this activity and have identified measures to control these risks and have implemented them.

Signature: _____ Date: _____

2. Gather the equipment needed for this exercise.

3. Read the following points and make sure that you perform them when wiring all circuits:

   - Write the IC pin numbers of the devices you work with on the circuit that you're instructed to wire.
   - Handle all ICs carefully - remember ESD can damage them.
   - When plugging hook-up wire into the breadboard to make connections to the IC's pins, use holes that are furthest away from the pin.

4. Wire the circuit of Figure 1 on the digital trainer ensuring that you connect the power supply to the IC. Pinout data for the 74LS151 can be found in Appendix 6.

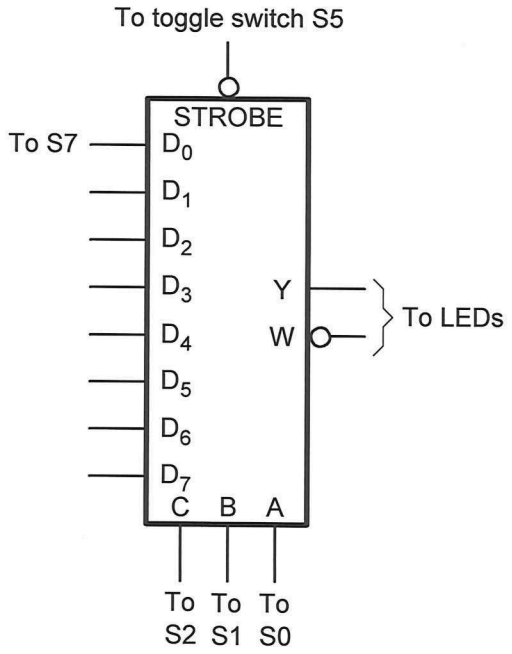


Figure 1

5. Turn on the trainer.

 Copyright: Not to be reproduced without permission

6. Set the switches connected the A, B and C inputs to logic-0.

7. Set the switch connected to the *STROBE* input to logic-0.

8. Throw the switch connected to the D_0 input between logic-0 and logic-1 and observe the outputs.

Question 1
What is the relationship between the D_0 input and the Y output?

_____

_____

Question 2
What is the relationship between the D_0 input and the W output?

_____

_____

9. Disconnect the switch to the D_0 input and connect it to the D_1 input.

10. Throw the switch connected to the input D_1 between logic-0 and logic-1 and observe the outputs.

11. Repeat Steps 9 and 10 for the remaining D inputs and observe the outputs.

Question 3
What is the relationship between the other D inputs (D_1 to D_7) and the Y and W outputs when the A, B and C inputs are logic-0?

_____

The teacher needs to check your work at this point...

12. Set the A input to logic-1 but keep the B and C inputs a logic-0.

13. Keep the *STROBE* input a logic-0.

14. Disconnect the switch to the D_7 input and connect it to the D_1 input.

15. Throw the switch connected to the D_1 input between logic-0 and logic-1 and observe the outputs.

Question 4
What is the relationship between the D_1 input and the Y output?

Question 5
What is the relationship between the D_1 input and the W output?

16. Disconnect the switch to the D_1 input and connect it to the D_0 input.

17. Throw the switch connected to the input D_0 between logic-0 and logic-1 and observe the outputs.

18. Repeat Steps 16 and 17 for the remaining D inputs and observe the outputs.

Question 6
What is the relationship between the other D inputs and the Y and W outputs when A is a logic-1 and B and C are logic-0?

Question 7
Based on the small set of tests that you've conducted so far, what is the relationship between the A, B and C inputs and the operation of the Y output?

Question 8
Based on your tests, what is the function of this IC?

The teacher needs to check your work at this point...

19. Return the *A* input to logic-0 and keep the *B* and *C* inputs a logic-0.

20. Disconnect the switch to the D_7 input and connect it to the D_0 input.

21. Throw the switch connected to the D_0 input between logic-0 and logic-1 and observe the outputs.

    Self check: If you've followed the instructions correctly, the *Y* output should be the same as the logic level on the D_0 input.

22. Set the *STROBE* input to a logic-1.

23. Throw the switch connected to the D_0 input again and observe the outputs.

Question 9
Based on your test, what happens to the device when the *STROBE* is a logic-1?

Question 10
What is another name for the *STROBE* input?

The teacher needs to check your work at this point...

24. Wire the circuit of Figure 2 on the trainer ensuring that you connect the power supply to the IC. Pinout data for the 74LS139 can be found in Appendix 6.

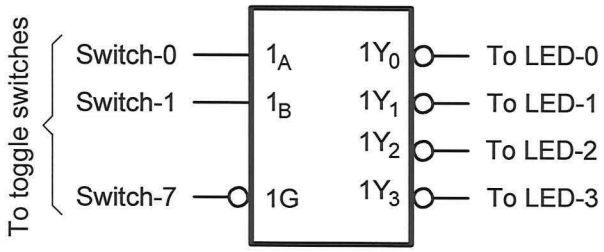


Figure 2

25. Turn on the trainer.

26. Set the switches connected the *A* and *B* inputs to logic-0.

27. Throw the switch-7 between logic-0 and logic-1 and observe the outputs.

Question 11
What is the relationship between the logic level on switch-7 and the outputs when the *A* and *B* inputs are logic-0?

28. Set the switches connected the *A* and *B* inputs so that *A* is logic-1 and *B* is logic-0.

29. Throw the switch-7 between logic-0 and logic-1 and observe the outputs.

Question 12
What is the relationship between the logic level on switch-7 and the outputs when *A* is a logic-1 and *B* is a logic-0?

30. Set the switches connected the *A* and *B* inputs so that *A* is logic-0 and *B* is logic-1.

31. Throw the switch-7 between logic-0 and logic-1 and observe the outputs.

Question 13
What is the relationship between the logic level on switch-7 and the outputs when *A* is a logic-0 and *B* is a logic-1?

_____

_____

32. Set the switches connected the *A* and *B* inputs to logic-1.

33. Throw the switch-7 between logic-0 and logic-1 and observe the outputs.

Question 14
What is the relationship between the logic level on switch-7 and the outputs when the *A* and *B* inputs are logic-1?

_____

_____

Question 15
Based on your tests, what is the function of this IC?

_____

The teacher needs to check your work at this point...

A clever application for multiplexer ICs

34. Wire the circuit of Figure 3 below ensuring that you connect the power supply to the 74LS151.

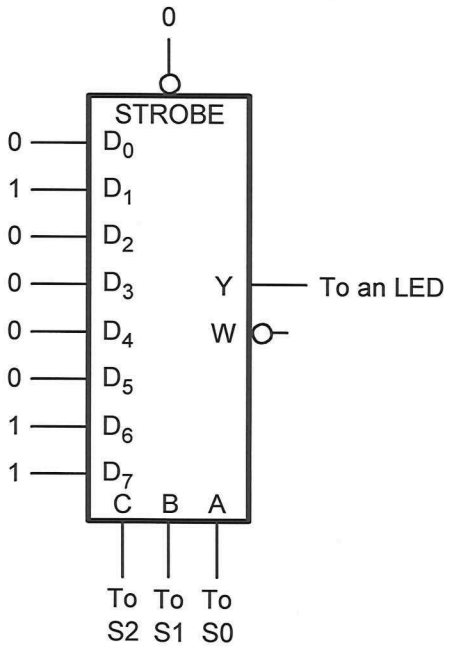


Figure 3

35. Set the *A*, *B* and *C* inputs to logic-0 and observe the Y output. Record your observations in the appropriate column of Table 1 on the next page.

36. Set the *A*, *B* and *C* inputs to the logic levels shown in the next row of Table 1.

37. Observe and record the Y output.

38. Repeat Steps 36 and 37 until Table 1 is complete.

 Copyright: Not to be reproduced without permission

Table 1

C	B	A	Z
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Question 16

As the data inputs to the multiplexer are hard-wired to the logic levels shown, they cannot change. That being the case, what is the multiplexer being used to simulate in this circuit? [Note: This application for a multiplexer is not discussed in the theory notes.]

Question 17

What change would you make to the circuit to have the multiplexer simulate a combinational logic circuit with a different truth table?

The teacher needs to check your work at this point...

Review questions

Answer these questions to check your understanding of what you have learnt from this chapter. Doing this will also help to prepare you for the assessments.

1. In terms of digital electronics, what is *time-division multiplexing*?

2. In terms of digital electronics, what is a *multiplexer*?

3. What is another word for a multiplexer?

4. In terms of digital electronics, what is a *demultiplexer*?

5. What is another word for a demultiplexer?

Questions 6 to 11 refer to the device shown in Figure 1 and its truth table

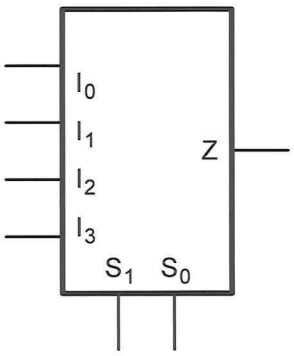


Figure 1

Inputs						Output
S_1	S_0	I_0	I_1	I_2	I_3	Z
0	0	0	×	×	×	0
0	0	1	×	×	×	1
0	1	×	0	×	×	0
0	1	×	1	×	×	1
1	0	×	×	0	×	0
1	0	×	×	1	×	1
1	1	×	×	×	0	0
1	1	×	×	×	1	1

6. The device is a

- ☐ 2-input mux.
- ☐ 2-input demux.
- ☐ 4-input mux.
- ☐ 4-input demux.

7. The *data* inputs are

- ☐ S_0 and S_1.
- ☐ I_0 to I_3.

8. The *select* inputs are

- ☐ S_0 and S_1.
- ☐ I_0 to I_3.

9. What is the relationship between the number on the S_0 and S_1 inputs and the data on the output?

10. What do the crosses in the truth table mean?

11. Which of the four truth tables below gives the same information as the truth table on the previous page but in a simpler way.

S_0	S_1	Z
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

S_1	S_0	Z
0	0	I_3
0	1	I_2
1	0	I_1
1	1	I_0

S_0	S_1	Z
0	0	I_3
0	1	I_2
1	0	I_1
1	1	I_0

S_1	S_0	Z
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Questions 12 to 16 refer to the device shown in Figure 2 and its truth table

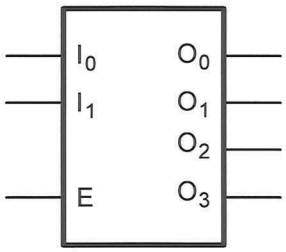


Figure 2

Inputs			Outputs			
E	I_1	I_0	O_0	O_1	O_2	O_3
0	×	×	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

12. The device is

- ☐ an encoder.
- ☐ a priority encoder.
- ☐ a decoder.
- ☐ a priority decoder.

13. The device can be operated as a

- ☐ 2-output mux.
- ☐ 2-output demux.
- ☐ 4-output mux.
- ☐ 4-output demux.

14. Data enters the device via

- ☐ the enable.
- ☐ I_0 only.
- ☐ I_1 only.
- ☐ either I_0 and I_1 depending on the logic level on the enable input.

15. The output that the input data gets copied to is determined by

- ☐ the enable.
- ☐ I_0 only.
- ☐ I_1 only.
- ☐ both I_0 and I_1.

16. What is the relationship between the number on the I_0 and I_1 inputs and which output the input data gets copied to?

Questions 17 to 22 refer to the 74LS153 IC. Data for the device can be found in Appendix 6

17. How many devices are inside the IC?

- ☐ 1
- ☐ 2
- ☐ 4
- ☐ 16

18. How many outputs does the device have?

- ☐ 1 (*Y*)
- ☐ 1 (*G*)
- ☐ 2 (*Y* and *G*)
- ☐ 4 (*C0*, *C1*, *C2* and *C3*)

19. How many select inputs does the device have?

- ☐ 2 (*A* and *B*)
- ☐ 4 (*C0*, *C1*, *C2* and *C3*)
- ☐ 6 (*A*, *B*, *C0*, *C1*, *C2* and *C3*)

20. How many data inputs does the device have?

- ☐ 2 (*A* and *B*)
- ☐ 4 (*C0*, *C1*, *C2* and *C3*)
- ☐ 6 (*A*, *B*, *C0*, *C1*, *C2* and *C3*)

21. According the device's function table, what kind of input is the *STROBE* input is acting as?

_____

22. The *STROBE* input is

- ☐ active low.
- ☐ active high.
- ☐ negative edge-triggered.
- ☐ positive edge-triggered.

Questions 23 to 28 refer to the 74LS154 IC. Data for the device can be found in Appendix 6

23. How many devices are inside the IC?

- ☐ 1
- ☐ 2
- ☐ 4
- ☐ 16

24. How many outputs does the device have?

- ☐ 1
- ☐ 2
- ☐ 4
- ☐ 16

25. According to the device's *function table*, under what conditions are all the outputs logic-1?

_____

_____

26. How many select inputs does the device have?

- ☐ 2 (*G1* and *G2*)
- ☐ 4 (*A*, *B*, *C* and *D*)
- ☐ 6 (*A*, *B*, *C*, *D*, *G1* and *G2*)

27. How many data inputs does the device have?

- ☐ 2 (*G1* and *G2*)
- ☐ 4 (*A*, *B*, *C* and *D*)
- ☐ 6 (*A*, *B*, *C*, *D*, *G1* and *G2*)

28. Data is entered into the device via

- ☐ either *G1* or *G2*.
- ☐ both *G1* and *G2*.
- ☐ both *G1* and *G2* for a logic-0 and either *G1* or *G2* for a logic-1.

Copyright: Not to be reproduced without permission

Student notes

Section 8 Analog to digital and digital to analog converters

Purpose To develop your skills in recognising and understanding the operation of digital to analog and analog to digital converters so that you will be able to test their operation.

Objectives At the end of this section you should be able to:

- Explain the need for analog to digital and digital to analog converters
- Define and explain the principle of analog to digital and digital to analog conversion
- List typical applications for ADCs and DACs
- Draw the schematic symbol for ADCs and DACs
- List and explain the basic performance characteristics of ADCs including: *range*, *resolution*, *conversion time* and *sampling rate*
- List the four ADC structures and their comparative advantages and disadvantages
- List and explain the basic performance characteristics of DACs including: *resolution*, *settling time* and *linearity*
- Interpret manufacturers' data for common commercially available 8-bit ADCs and DACs
- Explain *quantisation error* and its effect
- Describe the audible effect of *quantisation noise*
- Use an oscilloscope to test and measure DACs and ADCs

Introduction

For reasons discussed at the beginning of this unit, most new electronic communications systems are being implemented using digital electronics these days. However, humans are analogue creatures living in an analogue world. We generate analogue sounds when we speak, sing, play musical instruments, etc and our senses (sight, hearing, etc) respond to analogue stimuli (for example, we don't just simply hear or not hear sound, we're able to detect many levels of sound intensity). That being the case, analog to digital and digital to analog conversion are an essential part of digital communications systems meant for use by people. Obviously then, technicians need to know a little about them.

Analog to digital converters

Analog to digital converters (A/D converters or just ADCs) are typically used at the "front-end" of digital systems where analog information forms the input to the system. The mouthpiece of a mobile phone is an example. Mobile phone communications (and most telephony these days) are digital communications devices. But, when you speak into the mouthpiece, the microphone produces an analog electrical signal. So, one of the first circuits that this information must go through is an ADC. Another example where speech and other sounds are converted to digital is at the *mic* input on a computer's sound-card.

Importantly, speech and music are not the only analog input to digital systems. For example, heat is an analog input to temperature control systems such as air-conditioners, fridges, freezers, kilns, smelters, etc. Speed is an analog input to an engine management system of a car. And weight is an analog input to a set of digital scales. There are lots of examples.

Schematic symbol

The schematic symbol for an ADC is shown in Figure 1 below.

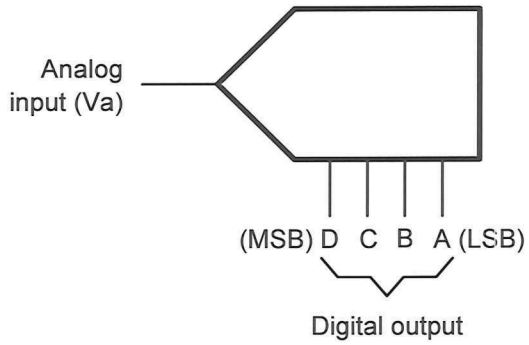


Figure 1 *The schematic symbol for analog to digital converters*

As you can see, the ADC in Figure 1 has an analog input and a 4-bit digital output. Four bits are used for this symbol purely for the purpose of example. Commercially available ADC ICs under $10 are usually 8-bit devices and there are many 10-bit and 12-bit devices available for under $30. Another important point that should also be made here is that, while the ADC in Figure 1 outputs parallel data, there are plenty of commercially available ADC ICs that output their data in serial form instead (in other words, the ICs are two devices in the one package, an ADC and a parallel to serial converter).

A definition for analog to digital conversion

Analog to digital conversion is the process of taking a sampled analog voltage or current and converting it to a binary number that is proportional to the size of the analog signal.

The use of the word "sampled" here needs discussing before going any further. Analog to digital conversion is not an instantaneous process - it takes a finite amount of time. Importantly, if the analog input changes during the conversion, the digital output will likely be incorrect. Unfortunately, the ADC has no control over the input voltage or current and cannot stop it from changing. So, ADCs usually sample the analog input voltage or current at a given moment and hold the sampled value for the duration of the conversion process. The circuit that does that is simply called a *sample-and-hold* circuit. Once the conversion is complete, the ADC can take another sample of the analog input for the next conversion.

Basic principle of operation

As the definition for analog to digital conversion implies, every sample of an analog input returns a number that corresponds to its relative size. To demonstrate this with an example, consider Table 1 below. It shows the digital output produced for a series of analog inputs to a hypothetical 3-bit ADC.

Table 1

Va to 3-bit ADC	Output of 3-bit ADC
0V	000
0.45V	001
0.9V	010
1.35V	011
1.8V	100
2.25V	101
2.7V	110
3.15V	111

Notice the "proportionality" between the inputs and outputs. That is, a doubling (or halving) of the analog input voltage returns a binary number that is also double (or half). For proof, consider a 0.9V analog input. A sample at this voltage returns the number 010 (or decimal 2) while a sample at twice the voltage (1.8V) returns the binary number 100 (or decimal 4). This satisfies our definition of analog to digital conversion. [Note: As an aside, there are non-linear ADCs where the input and output are not proportional in this way. However, the non-linearity isn't random. Instead, the ADC's performance corresponds to a predetermined non-linear mathematical expression. As this is an introduction to the topic these more specialised devices are not discussed here.]

Now, consider the information in Table 1 more closely. What do you think happens when the sampled input voltage to our hypothetical ADC is different to ones specified? The answer to this question is important to understanding a serious issue in the whole process of analog to digital to analog conversion. Answer: The ADC outputs a number corresponding to the voltage in Table 1 that is closest to the sampled input voltage. For example, if the sample is 1V, the nearest value in Table 1 is 0.9V so the ADC outputs the code 010.

All ADCs operate this way and this has two important implications. First, any set of specified analog inputs like that in Table 1 must actually be *centre values*. That is, each specified voltage is in the centre of a range of voltages that will output the same number. These are also known as *quantisation levels*. Second, and more importantly, as there are many potential samples that can return the same number this must mean that analog to digital conversion is inherently wrong most of the time! To explain, 1V is 100mV bigger than 0.9V but our hypothetical ADC will output the code 010. Thereafter, the record of that sample is 100mV wrong. This problem is known as *quantisation error*.

Without getting into the detail of how digital to analog converters (DACs) work, quantisation error causes a problem called *quantisation noise* when the data is converted back to analog. And, the greater the quantisation error, the greater the quantisation noise. This issue is discussed more towards the end of the chapter.

ADC performance characteristics

Common performance characteristics of ADCs for important comparisons between them include:

- Range
- Resolution
- Conversion time
- Sampling rate

Range

As its name implies the *range* of an ADC specifies the minimum and maximum voltages or currents that the device can convert to binary numbers. Some ADCs are bipolar which means that the minimum voltage is a negative value while the maximum voltage is positive. For example a bipolar 8-bit ADC with a range of ±2V converts -2V to 00000000 and +2V to 11111111. Other devices are unipolar and so may range from say 0V to 5V. For example, a unipolar 8-bit ADC with a range of 0V to 5V converts 0V to 00000000 and 5V to 11111111.

More sophisticated ADC ICs allow designers to "program" the range which is a very handy feature. Setting the range to just above the likely range of your analog input allows for maximum resolution.

Resolution

The *resolution* of an ADC is an indication of the number steps between the minimum and maximum values in the ADC's range that produce a unique number. To explain this a little, consider the hypothetical ADCs in Figures 2a and 2b below both having a range of 0V to 3.15V.

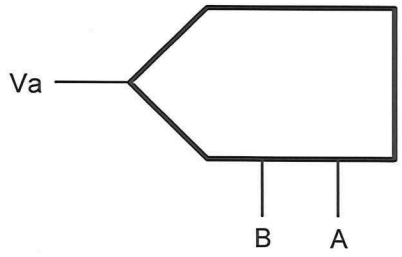


Figure 2a *A 2-bit ADC*

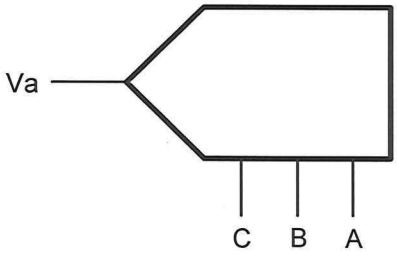


Figure 2b *A 3-bit ADC*

For the centre values shown in Table 2 below, the ADCs would produce the numbers shown.

Table 2

Va to 2-bit ADC	Output of 2-bit ADC	Va to 3-bit ADC	Output of 3-bit ADC
0V	00	0V	000
		0.45V	001
1.05V	01	0.9V	010
		1.35V	011
2.1V	10	1.8V	100
		2.25V	101
3.15V	11	2.7V	110
		3.15V	111

As you can see, the 3-bit ADC outputs more numbers over the same range. This means that the 3-bit ADC can "resolve" (or distinguish between) smaller portions of the same input than the 2-bit ADC. In other words, its resolution is better.

ADC resolution has obvious implications for quantisation error (and hence quantisation noise). To explain, suppose a sampled analog input to our two ADCs is 800mV. This figure is 205mV away from the nearest centre value of the 2-bit ADC but only 100mV from the nearest centre value of the 3-bit ADC. In other words, the quantisation error is smaller for the device with the better resolution and this will be true for most samples across the range.

To conveniently compare the resolution of ADCs, it needs to be quantified. One method of doing so simply involves quoting the number of bits on the ADC's output. This makes sense because we know that a 4-bit ADC has a better resolution than a 3-bit ADC. That being the case, logically, a 5-bit ADC must be better than a 4-bit ADC but not as good as a 6-bit ADC and so on.

Another method of quantifying ADC resolution involves using the number of an ADC's output bits to calculate a percentage figure (which tells you the greatest change in input voltage - as a percentage of the difference in the range - that is needed to cause a change in the number on the ADC's output). An equation that lets you do that is:

$$Resolution = \frac{1}{2^n - 1} \times 100$$

Where: n is the number of outputs

Applying this equation to our ADCs in Figures 2a and 2b we find it gives their resolutions as:

$$Res = \frac{1}{2^n - 1} \times 100 \qquad Res = \frac{1}{2^n - 1} \times 100$$

$$Res = \frac{1}{2^2 - 1} \times 100 \qquad Res = \frac{1}{2^3 - 1} \times 100$$

$$Res = 33.33\% \qquad Res = 14.29\%$$

As you can see from these two calculations, the smaller the percentage, the better.

Conversion time

The *conversion time* is the time required by an ADC to sample the analog input, convert it to a digital value, then update its outputs. Figures for ADCs generally range from milliseconds to nanoseconds depending on the ADC architecture (more later).

Sampling rate

The *sampling rate* of an ADC is the maximum number of samples of the analog input that the ADC can take per second. The sampling rate is usually expressed simply in samples per second (Sa/s or SPS) or, less often, in Hertz (Hz). So, an ADC that can sample the analog input ten thousand times per second has a sample rate of 10kSa/s, 10kSPS or 10kHz.

The sample rate is obviously affected, in part, by the ADC's conversion time. The quicker the conversion time, the sooner the device can take its next sample.

Depending on the analog input, the sampling rate can be a critical ADC characteristic. It determines the maximum frequency of an AC input that can be converted to digital. Without going into the specifics of why, the sampling rate must be at least twice the highest frequency component in the AC input (a figure known as the *Nyquist Sample Rate* and named after its discoverer). For example, let's consider digitally recorded music. For high fidelity, we'd want to capture everything that we can hear which is the spectrum between about 20Hz and 20kHz. That being the case, the sample rate must be at least forty thousand times per second. In fact, the sampling frequency for digitally mastered music is 44.1kHz which is slightly higher than the minimum.

ADC structures - advantages and disadvantage

There are a number of different methods for implementing ADCs and these are known as *structures*. When ADCs were implemented using a combination of discrete components and ICs (like counters and op amps) it was necessary to explain how the different structures operated because individual parts that failed could be replaced. These days however, you're much more likely to encounter ADCs as dedicated ICs so asking you to learn how these structures operate is a bit hard to justify. That said, you are still expected to know the names of some common ADC structures and their comparative advantages and disadvantages.

Four of the ADC structures include:

- Digital ramp converter

  This ADC suffers from having a variable conversion time. The bigger the analog input voltage the longer the conversion time. This can be a problem because variable timing usually requires the digital circuitry they interact with to be more complicated. Another disadvantage of the digital ramp is the effect of noise on the conversion. All signals have some amount of electrical noise superimposed on them that can cause spurious conversions. For example, suppose that the voltage of a signal at a particular sampling instant is 2V. But suppose also that the noise that is superimposed on the signal at that instant peaks at +0.1V. This would make the sampled voltage 2.1V instead of 2V and so the wrong number may be generated (depending on the ADC's resolution).

- Successive approximation converter

  This ADC has a fixed conversion time that is relatively quick and so this is the most widely used ADC structure. That said, it suffers from the same noise problems as the digital ramp converter.

- Dual slope converter

  This ADC has excellent noise immunity and so is preferred when the accuracy of the conversion is critical (such as in test equipment like digital multimeters). However, the conversion time is variable and is slow (typically 10ms to 100ms) so this is a significant limitation.

- Flash converter

  As its name implies, this is the fastest of the ADCs and so is useful when converting high frequency baseband signals like video or the inputs to a digital CRO. However, it suffers from the same noise sensitivity as the digital ramp and successive approximation converters.

For an insight into the advantages and disadvantages of these ADC structures, Appendix 4 explains their operation. Remember though, you'll not be tested on this.

Integrated circuit ADCs

As analog to digital conversion is so important to the operation of digital systems, there's probably at least a thousand commercially available ADC ICs. To give you a feel for these devices, we'll consider just one: the ADC0804.

The ADC0804

Specifications and features of this device include:

- Supply: 4.5V to 6.3V (so can have a 5V TTL Vcc)
- Range: 0V to Vcc
- Resolution: 8-bit (or 0.392%)
- Total error: ±1 LSB
- Conversion time: 100µs
- Successive approximation conversion used
- Digital outputs that are both TTL and MOS compatible
- On chip clock (so no external clock source needed)
- Can be operated as a stand-alone (or *free-running*) ADC
- Control inputs (and tri-state data outputs) provided for interfacing to microprocessors
- Available in DIL and SOIC packages

The minimum circuit requirements to operate the ADC0804 as a stand-alone TTL ADC are shown in Figure 3 below.

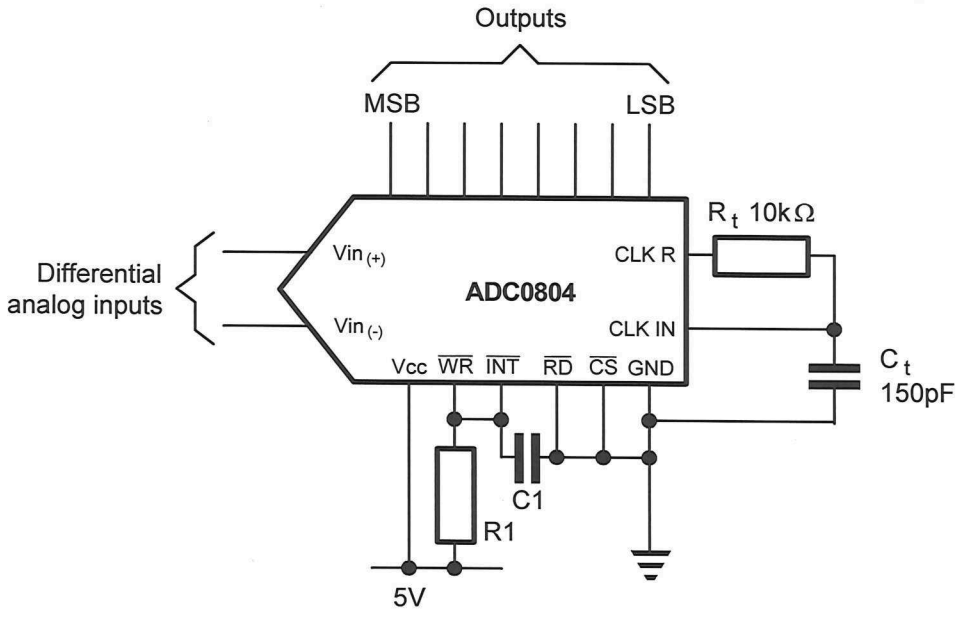


Figure 3 *The minimum operating requirements for the ADC0804 as a stand-alone TTL ADC*

There are a few interesting points to make about the arrangement in Figure 3. First, notice that the ADC0804 has two analog inputs. These are differential inputs like the inputs to an op amp. (In fact, notice that they're labelled in the same way as op amp inputs.) This increases the device's flexibility (and hence the possibility for greater sales) as it allows it to convert both balanced and unbalanced analog signals. For unbalanced inputs, simply connect $Vin_{(-)}$ to common and the analog signal to $Vin_{(+)}$. Reversing these connections effectively inverts the analog input signal which could be handy is some situations.

Second, notice that the IC has the following *control* inputs: $\overline{WR}$, $\overline{INTR}$, $\overline{RD}$ and $\overline{CS}$. These inputs allow a microprocessor to control when the ADC takes its sample and when it outputs its data. To operate the ADC as a stand-alone device (as in Figure 3), the $\overline{RD}$ and $\overline{CS}$ inputs must be hard-wired to logic-0 while the $\overline{WR}$ and $\overline{INTR}$ inputs must be hard-wired to logic-1. That said, the latter inputs must be pulsed low on power-up to get the ADC going. So, a resistor and capacitor arrangement is used (see *R1* and *C1*) to hold these pins low initially then pull them high as the capacitor charges. [Note: This RC arrangement was introduced to you in the section on D-type flip flops.]

Third, although the IC has an in-built oscillator, flexibility is provided by allowing the user to set its oscillating frequency. This is done using the resistor and capacitor (R_t and C_t) connected to the *CLK R* and *CLK IN* inputs. For the component values shown, the clock frequency is 606kHz (calculated using the equation $f = \frac{1}{1.1RC}$ given in the manufacturer's data).

Digital to analog converters

Digital to analog converters (D/A converters or just DACs) are typically used at the "back-end" of digital systems where information must be output in analog form. For an example, let's return to the mobile phone. The mobile phone receives the digital information and if the data is part of a voice communication it must be sent to a DAC to convert it to analog before being amplified and sent to the speaker that you listen to. Another example where speech and other sounds are converted from digital back to analog is at the speaker output on a computer's sound-card and the output circuitry of a dedicated network audio player (or "music streamer").

Again, speech and music are not the only types of analog output from digital systems. Digital control applications convert digital information to analog to determine things like the speed or position of a motor, etc.

Schematic symbol

The schematic symbol for a DAC is shown in Figure 4 below.

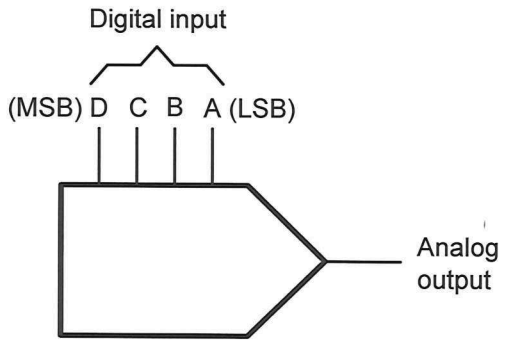


Figure 4 *The schematic symbol for digital to analog converters*

As you can see, the DAC in Figure 4 has a 4-bit digital input. Four bits are used for this symbol purely for the purpose of example. Commercially available DAC ICs under $10 are usually 8-bit devices and you don't have to pay too much more for 10-bit and 12-bit devices. Another important point that should also be made here is that DACs with serial data inputs are also readily available.

A definition for digital to analog conversion
Digital to analog conversion is the process of taking a binary number and converting it to a voltage or current that is (usually) proportional to the number.

Principle of operation
As the definition for digital to analog conversion implies, the basic principle of operation of DACs is fairly straight forward. Every unique binary number into the device causes it to output a unique voltage or current. For example, a hypothetical 3-bit DAC outputs the voltages shown in Table 3 below for the specified binary numbers on its input.

Table 3

Binary input	Analog output
000	0V
001	0.5V
010	1V
011	1.5V
100	2V
101	2.5V
110	3V
111	3.5V

Notice that the DAC's output is proportional to its binary input. That is, every doubling (or halving) of the binary number results in double (or half) the output voltage. For example, 010 (which is decimal 2) causes the DAC to output 1V and 100 (which is decimal 4) causes the DAC to output 2V. [Note: As some ADCs have a mathematically defined non-linear conversion process, DACs with the inverse non-linear relationship are also available. As this is an introduction to the topic, these specialised devices are not discussed here.]

DAC performance characteristics

Common performance characteristics of DACs for important comparisons between them include:

- Resolution
- Settling time
- Linearity

Resolution

The *resolution* of a DAC can be quoted in terms of the number of bits on its input or calculated using the equation:

$$\%\ Resolution = \frac{1}{2^n - 1} \times 100$$

Where "n" is the number of inputs

A DAC's resolution should match the resolution of the ADC that generated the digital data in the first instance. If the DAC's resolution is lower than the ADC's resolution, quantisation noise on the DAC's output increases. If the DAC's resolution is higher than the ADC's, you've wasted your money because you don't get any improvement in system performance.

Settling time

The *settling time* is the time it takes for the DAC's output voltage to change and settle in response to a change of input. This characteristic is analogous to the propagation delay of a logic gate.

Linearity

The *linearity* of a DAC describes how much the output deviates from a straight line for a digital ramp output. Figure 5a shows a linear output ramp and 5b shows a ramp with an exaggerated non-linearity (the dashed line shows what the output should have been like).

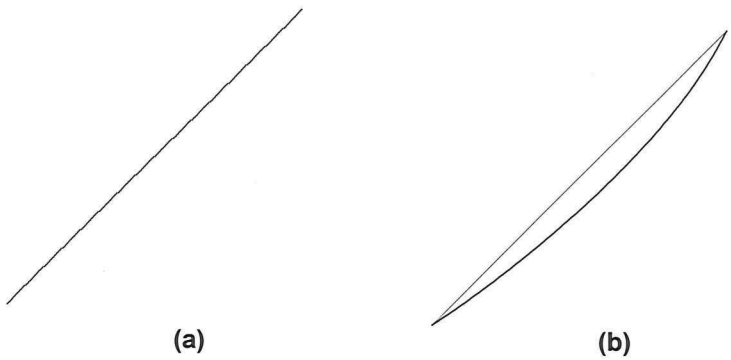


Figure 5 *a) A linear output response; and b) an exaggerated non-linear output response*

DAC types

The simplest of the DAC types is the summing amplifier that you learnt about earlier in the course. However, this circuit suffers from several problems and is seldom used commercially (if at all). A more popular design is the R/2R ladder network DAC.

However, as with ADCs, you're unlikely to encounter discrete component DACs these days as there are so many fully integrated DAC ICs. That being the case, it's hard to justify asking you to learn how these circuits work. If you're interested, explanations are provided in Appendix 5.

Integrated circuit DACs

Like analog to digital conversion, digital to analog conversion is so important to the operation of digital systems that there's probably at least a thousand commercially available DAC ICs. To give you a feel for these devices, we'll consider just one: the DAC08CP.

The DAC08

Specifications and features of this device include:

- Supply: ±4.5 to ±18V
- Resolution: 8-bit (or 0.329%)
- Settling time: 100ns
- Non-linearity: 0.1%
- Digital inputs are TTL, CMOS, ECL, HTL and PMOS compatible
- Complementary current outputs
- Outputs are current sinking
- Available in DIL and SOIC packages

The minimum circuit requirements to operate the DAC08 are shown in Figure 6 on the next page.

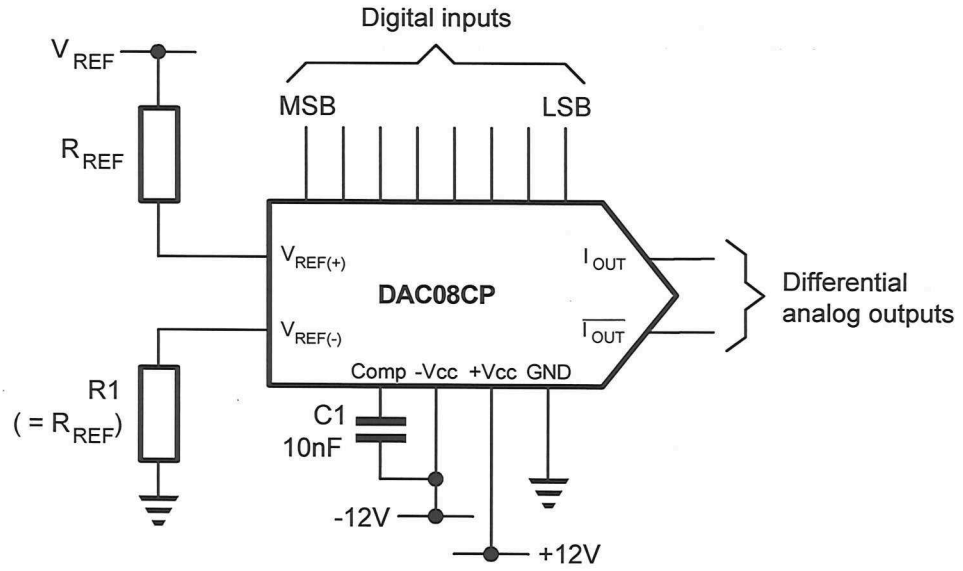


Figure 6 *Typical connections for a DAC08CP IC*

There are a couple of interesting points to make about the arrangement in Figure 6. First, notice that the DAC08 has two analog outputs. These are differential outputs that allow for the device to provide balanced or unbalanced analog outputs. Second, the device has current outputs instead of voltage outputs. This further increases the device's flexibility. If output voltages are required, the outputs are simply connected to resistors as shown in Figure 7 below (for an unbalanced unipolar negative output).

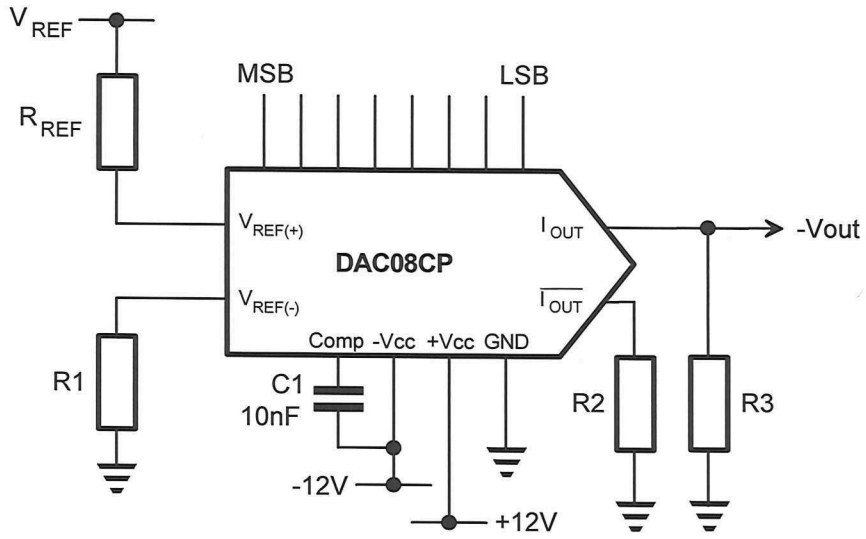


Figure 7 *The DAC08CP wired to provide an unbalanced unipolar negative output voltage*

Quantisation noise

The issue of quantisation error has been explained earlier in this section. Recall that this is the error inherent in all ADCs due to the fact that the binary numbers it outputs represents a range of analog input voltages. Quantisation error is responsible for a problem called quantisation noise on the output of DACs and this issue can be discussed in a little more detail now that DACs have been explained.

Consider the analog to digital then digital to analog arrangement in Figure 9 below. The number of bits for the binary number (that is, the resolution of the arrangement) isn't specified so we can consider a few examples.

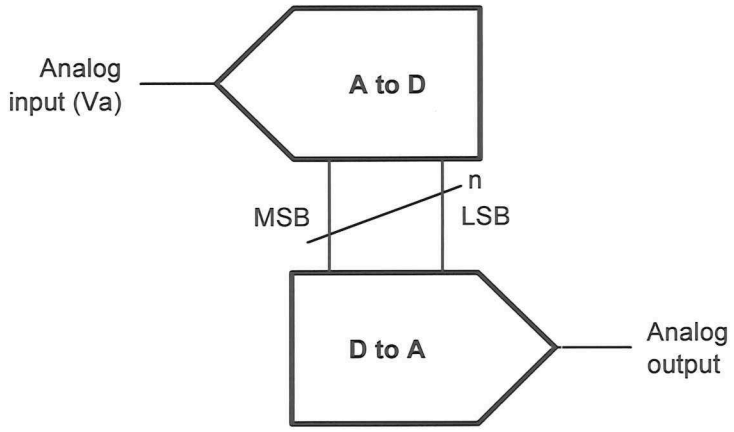


Figure 9 *An analog to digital then digital to analog arrangement with a non-specific resolution*

Now suppose that an analog ramp voltage is connected to the ADC's input that changes across the device's range. Such an input is shown in Figure 10 below.

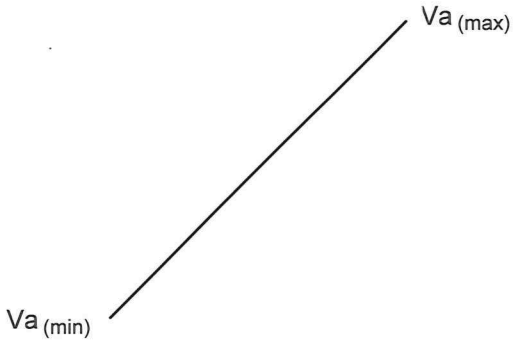


Figure 10 *An analog ramp voltage is connected to the input of the ADC*

As the ramp changes across the ADC's range, the device outputs every code from zero to full-scale. For example, if the ADC has a 2-bit resolution (or 33.33%) it outputs the numbers 00, 01, 10 then 11. If the ADC has a 4-bit resolution (or 6.67%) it outputs the numbers 0000, 0001 0010 and so on up to 1111. And, if the ADC has an 8-bit resolution (or 0.39%) it outputs the numbers 00000000, 00000001, 00000010 and so on up to 11111111.

The DAC dutifully converts these numbers to analog voltages and Figure 11 below shows what we would see on the outputs of the DAC for 2-bit, 4-bit and 8-bit systems.

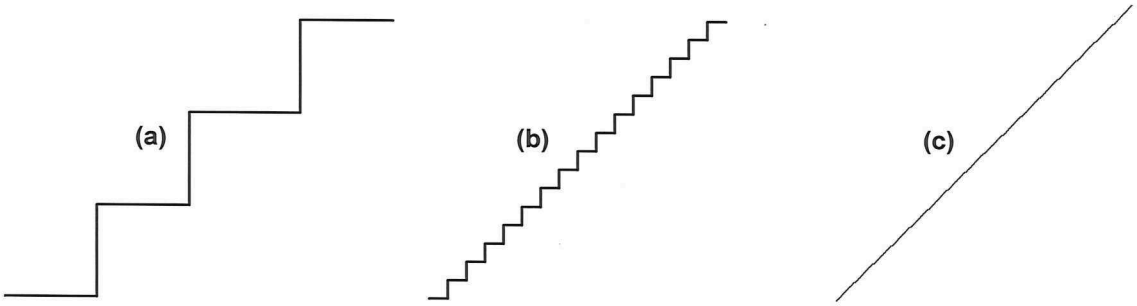


Figure 11 *The reconstructed ramp out of: (a) a 2-bit system; (b) a 4-bit system; and (c) an 8-bit system*

Clearly, the greater the resolution of the device, the more closely the DAC's output resembles the original analog input. This is obviously important because we usually want the reconstructed signal to be as close to the original as possible, especially when the original is speech, music or video.

The question now is: Why is the difference between the DAC's output and the original analog input referred to as quantisation noise? Why not just call it quantisation distortion? The answer: If you listen to a signal with an audible level of this error it sounds exactly like the classic hiss of white noise. (For an example of this, do a search on Wikipedia for "quantisation error" then scroll down to the button that plays a short piece of music with a slowly increasing level of quantisation noise.)

Importantly, if the output of the DAC is examined using a spectrum analyser you'll find all the sinewaves of the original analog signal peaking above a sea of white noise (which has an even distribution of sinewaves across the entire radio frequency spectrum each having the same power density on average).

As mentioned above, the greater the resolution of the system the more closely the DAC's output resembles the original analog input. But, we can put this another way now. The greater the resolution, the lower the level of quantisation noise that has been added.

Can quantisation noise be eliminated? No, increasing the resolution only ever reduces it. Filtering also helps so DACs are often followed by a low-pass filter (often referred to as an integrator) to clean the signal up. That said, there's little benefit in reducing the quantisation noise below the level of background noise in the original analog signal.

Copyright: Not to be reproduced without permission

Student notes

Skill practice 8

Practise testing and measuring analog to digital and digital to analog converters

This exercise is practise for the sorts of skills you may be required to perform in a practical test. Remember, in any practical tests you will be working alone so make sure that you can perform all the steps. It should take you about 1¼ hours to complete this exercise.

Equipment

- Emona Telecoms Trainer (also known as the "Biskit")
- DMM
- two Emona Telecoms Trainer CRO leads
- 2mm banana leads

Remember:

Follow TAFE NSW WHS guidance at all times!

Work tasks

1. Read your WHS responsibilities at the top of the form below. Then conduct a WHS risk assessment and record your findings in the space provided.

Responsibilities of students under the Model WHS Act: s28

- Take reasonable care for your own health and safety by working safely at all times
- Take reasonable care to ensure that your acts or omissions don't put the health and safety of others at risk
- Follow all TAFE NSW WHS guidance and comply with all reasonable instructions from TAFE NSW staff to assist them in complying with the TAFE NSW WHS requirements
- In addition to the above, you must:
  - use and maintain machinery, tools and all other equipment properly and safely
  - ensure that your work area is free of hazards
  - notify a TAFE NSW staff member of actual or potential hazards
  - wear/use prescribed safety equipment
  - take notice of any safety signs and adhere to their instructions

Risks involved in this activity include:	**Control measures:**
■ Trip hazards (eg students bags) ■ Objects dropped on feet (while equipment is taken to and from benches).	■ Move bags and other objects from walkways ■ Plan lifting of equipment
Others: _____ _____ _____	Other: _____ _____ _____

My signature here indicates that I have read and understand my responsibilities under the Model WHS Act s28 (detailed above). I have also conducted a risk assessment before undertaking this activity and have identified measures to control these risks and have implemented them.

Signature: _____ Date: _____

Part A - Analog to digital conversion (DC input)

2. Gather the equipment needed for this exercise.

3. Before going any further, read the following notes.

Notes:
Part A of this practical exercise gets you to implement the 8-bit ADC shown in Figure 1 below using the PCM Encoder module on the Emona Telecoms Trainer.

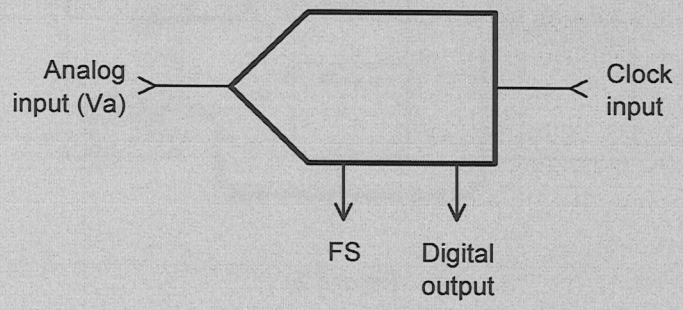


Figure 1

Notice that the schematic symbol shows that the ADC has only one data output. This is because the device continuously outputs serial data instead of parallel. Conveniently, this will allow you to examine the output data using an oscilloscope. [Note: If the output data was parallel, we would need either an 8-channel CRO to view all of the bits at once or a special and expensive piece of digital test equipment called a *logic analyser*.]

For your information, this ADC outputs the data word's most significant bit (bit-7) first and the least significant bit (bit-0) last.

Notice that the ADC also has an active high output labelled "FS". This stands for *frame synchronisation* and the output is asserted at the same time that the ADC outputs bit-0 of the 8-bit serial data word. This can be used by a PCM Decoder to identify the first bit of each new data word. You'll use it for the same purpose when viewing the data word on the CRO.

4. Turn on the Emona Telecoms Trainer.

5. Locate the PCM Encoder module and set its *PCM/TDM* slide switch to the *PCM* position.

6. Connect the set-up shown in Figure 2 below using the 2mm banana-plug leads.

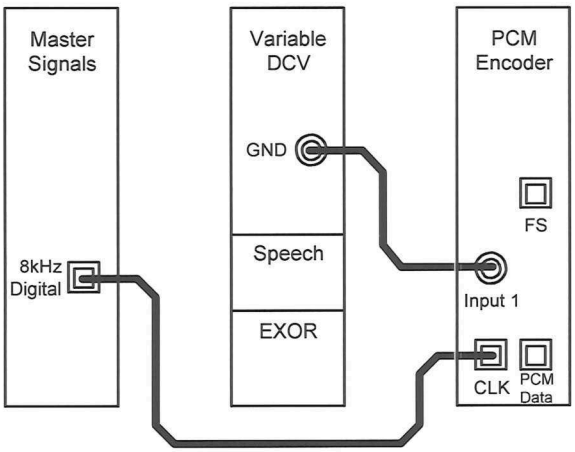


Figure 2

Note: This applies an 8kHz squarewave to the ADC's clock input and 0V DC to its analog input as shown in Figure 3 below.

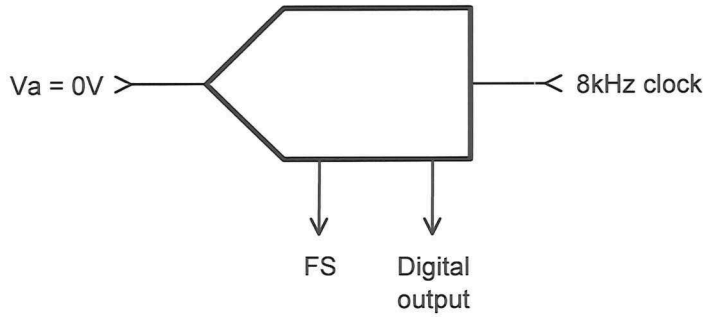


Figure 3

7. Set the *Input Coupling* controls of both of the CRO's channels to the *DC* position.

   Note: This is a usual setting for the coupling control of CROs when looking at digital signals.

8. Connect the CRO's Channel-1 input to the PCM Encoder module's *FS* output and adjust the CRO to display the FS signal.

   Note: Connect the CRO lead's black banana plug to one of the ground (*GND*) sockets in the middle of the trainer.

9. Adjust the CRO's controls listed below as indicated:

   - Set the *Slope* control (also sometimes labelled *Sync*) to the "-" position to trigger the CRO on the *FS* pulse's negative edge.
   - Adjust the *Horizontal Position* so that the trace begins at exactly the beginning of the graticule on its left end.
   - Set the *Horizontal Sweep* to 0.1ms/div (or 100µs/div).
   - Unlock the *Variable Horizontal Sweep* control (the blue or red knob inside the *Horizontal Sweep's* control) and adjust it until the pulse in the FS signal is in the same position as shown in Figure 4.

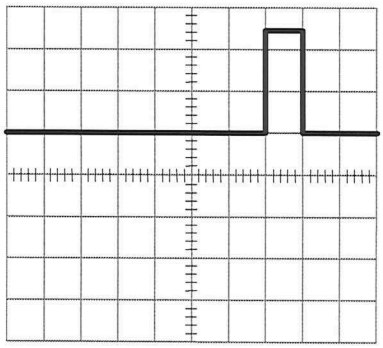


Figure 4

Note: The adjustments you have just made set up the CRO so that each bit on the PCM Encoder module's output is exactly one division wide (like the *FS's* pulse is).

The teacher needs to check your work at this point...

10. Draw the *FS* signal to scale in the upper third of the graph below.

    Note: Leave enough room below it for two more digital signals.

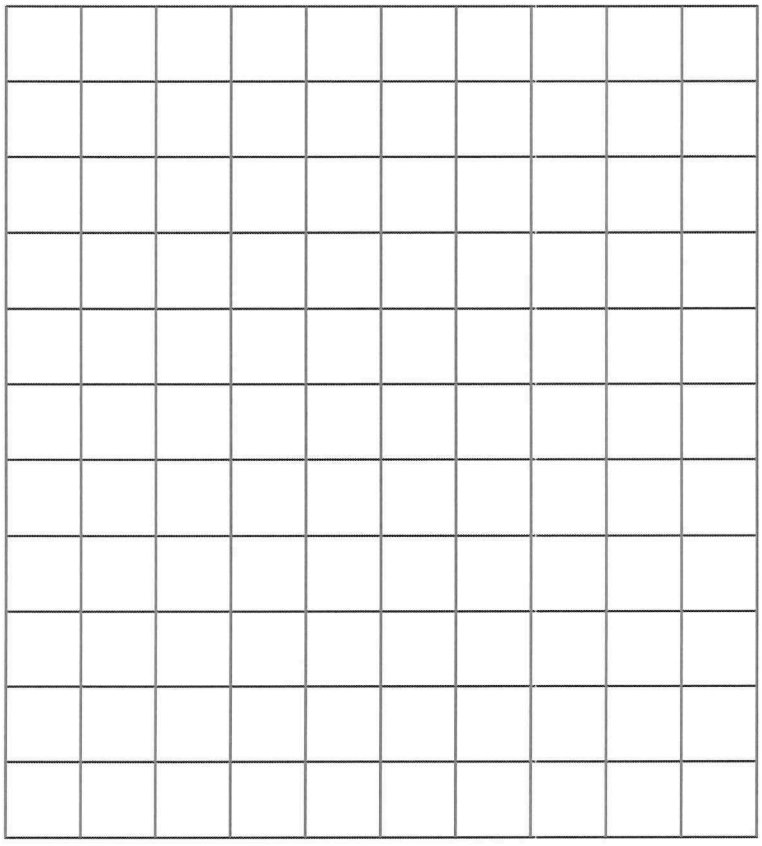
11. Connect the CRO's Channel-2 input to the PCM Encoder module's *CLK* input.

12. Adjust the CRO to view both signals together **but don't** change any of the CRO controls that you adjusted for Step 8.

13. Draw the clock signal to scale in the middle of the graph.

14. Connect the CRO's Channel-2 input to the PCM Encoder module's *PCM DATA* output.

15. Draw the PCM Encoder module's *PCM DATA* output to scale at the bottom of the graph.



The teacher needs to check your work at this point...

Question 1
According to the notes on page 8-22, what is the length of each data word on the PCM Encoder module's output?

Question 2
According to the notes associated with Step 8, how wide is each of the bits on the CRO's display?

Question 3
Given your answer to Question 2, how many bits of data is the CRO displaying?

Question 4
Given your answer to Question 3, what are the extra bits? **Tip:** The notes on page 8-22 will help you with this question.

The teacher needs to check your work at this point...

Question 5
Indicate which bit on the PCM Encoder module's output is the LSB (bit-0) of the 8-bit data word. **Tip:** Again, the notes on page 8-22 will help you answer this question.

Question 6
Indicate which bit on the PCM Encoder module's output is the MSB (bit-7) of the 8-bit data word. **Tip:** As above.

 Copyright: Not to be reproduced without permission

Question 7
Now that you have identified the start and end of the data word on the PCM Encoder module's output, what binary number is the analog input of 0V being converted to?

_____

Question 8
Why do you think that 0V converted to the number in your answer above and not 00000000?

_____

_____

The teacher needs to check your work at this point...

16. Modify the set-up so that the PCM Encoder's analog input (*INPUT-1*) is connected to the Variable DCV module's *VDC* output as shown in Figure 5 below.

Note: Thin lines show leads already in place that must not be changed.

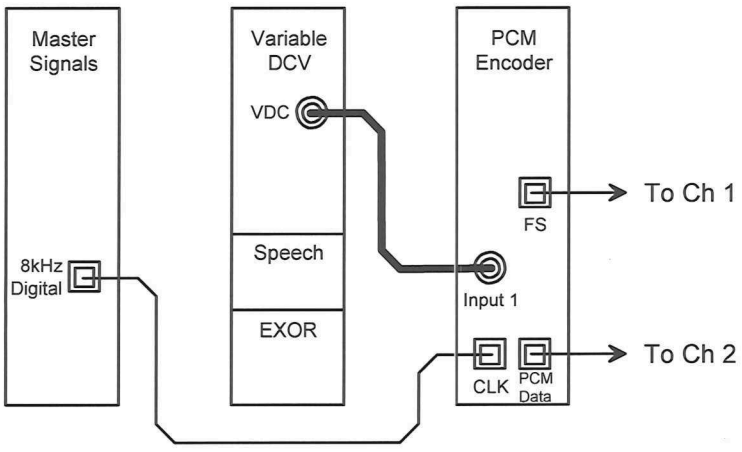


Figure 5

17. Vary the Variable DCV module's *VDC* control and observe the effect on the PCM Encoder module's *PCM Data* output.

Question 9
Why does the PCM Encoder module's output data change as you perform Step 16?

_____

_____

18. Verify your answer to the question above by using the DMM to monitor the Variable DCV module's *VDC* output while you repeat Step 16.

19. Use the DMM to determine the PCM Encoder module's *range*. **Tip:** If you're not sure what the range specification is, read the notes on page 8-4.

The PCM Encoder module's range is: _____

The teacher needs to check your work at this point...

Part B - Analog to digital conversion (AC input)

This part of the experiment lets you convert a 2kHz sinewave to digital. However, before you do, there's an important fact about the PCM Encoder module you should know. Although the encoder's clock is currently 8kHz, the ADC's actual sampling rate is much lower. To explain, once the analog-to-digital conversion has been made, the parallel data word must be converted to serial form. Recall from earlier work that parallel-to-serial conversion typically takes one clock pulse per bit to complete. As the PCM Encoder module is an 8-bit ADC it must take eight clock pulses to convert each number to serial. That being the case, there is no point in the PCM Encoder module sampling the analog input at a rate higher than $\frac{1}{8}$ of the clock frequency (which is 1kHz or 1kSa/s).

Importantly, a sampling rate of 1kHz is insufficient for the next part of the practical exercise. So, one of the modifications to the circuit involves increasing the clock frequency.

20. Modify the set-up as shown in Figure 6 below.

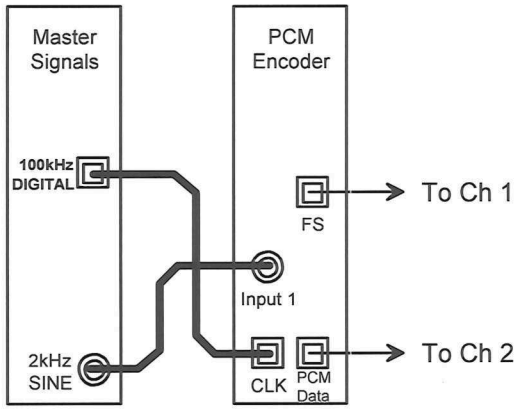


Figure 6

21. Adjust the CRO's *Horizontal Sweep* control to 20µs/div.

22. Return the CRO's *Variable Horizontal Sweep* control to the detent (that is, the locked) position.

    Note: The *FS* signal should now be a steady display similar to before.

23. Adjust the CRO's triggering controls to see if you can get the digital data on the PCM Encoder module's output to stabilise on the display.

Question 10
Why can't you get the data on the PCM Encoder module's output to stabilise on the CRO's display?

Question 11
Why is the data continuously changing?

Question 12
Why was it necessary to increase the PCM Encoder module's clock frequency to at least 32kHz? **Tip:** If you're not sure, read the notes on sampling rate on page 8-7 **and** the notes at the top of the previous page.

The teacher needs to check your work at this point...

Part C - Digital to analog conversion

This part of the practical exercise gets you to convert the digital data currently on the PCM Encoder module's output back to analog using the PCM Decoder module. The entire set-up can be represented by the block diagram in Figure 7 below.

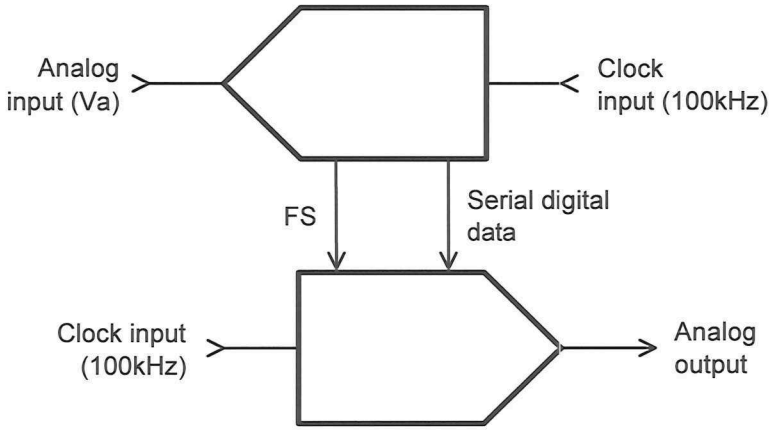


Figure 7

24. Connect the set-up shown in Figure 8 below.

Note: Remember, thin lines show leads already in place that must not be changed.

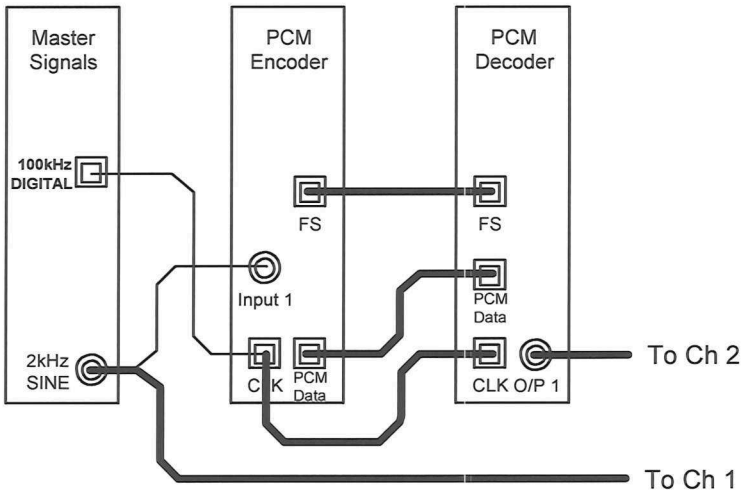


Figure 8

25. Adjust the CRO to show the original 2kHz sinewave on Channel-1 and the PCM Decoder module's output on Channel-2.

26. Compare the visible difference between the signals.

Question 13
What is the name used to describe the fact that the output of the PCM Decoder module isn't a perfect copy of the original on the PCM Encoder module's input?

_____

Question 14
What causes this problem and where does it occur?

_____

_____

Question 15
When audible, what does this problem sound like?

_____

Question 16
What are two ways of reducing this problem?

_____

_____

_____

The teacher needs to check your work at this point...

27. To verify one of your answers to Question 16, locate the Tuneable Low-pass Filter module (top-right) and turn its *GAIN* control to about the middle of its travel.

28. Increase the Tuneable LPF module's cut-off frequency to maximum by turning its f_C control fully clockwise.

29. Modify the set-up as shown in Figure 9 below.

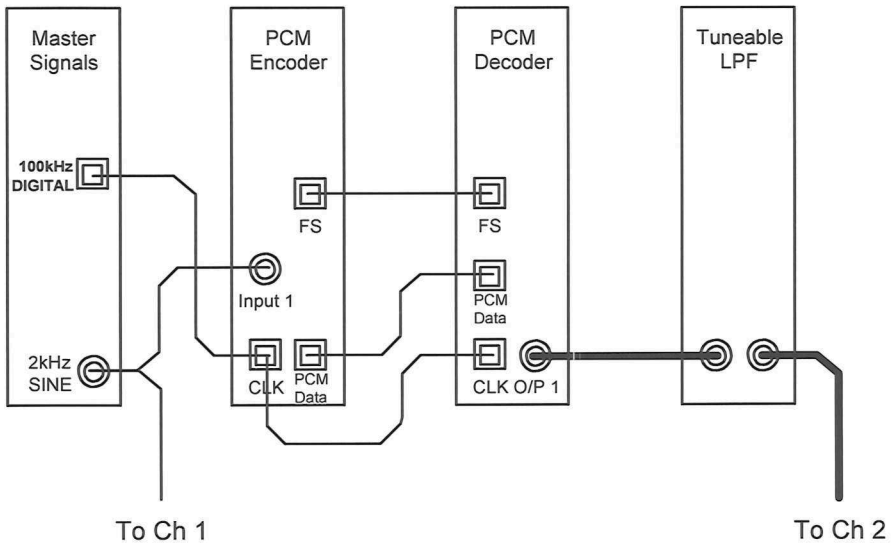


Figure 9

30. Slowly reduce the Tuneable LPF module's cut-off frequency to reduce its pass-band and stop when its output is a clean copy of the original 2kHz sinewave (ignoring any phase shift).

Question 17
What has the filter removed from the PCM Decoder module's output?

_____

_____

The teacher needs to check your work at this point...

Review questions

Answer these questions to check your understanding of what you have learnt from this chapter. Doing this will also help to prepare you for the assessments.

1. Define analog to digital conversion.

2. Draw the schematic symbol for an 8-bit DAC and label all inputs and outputs.

3. What does the *range* figure of an ADC tell us about the device?

4. Why must an ADC *sample* its analog input voltage before converting?

5. Two ADCs are compared and one has a better resolution than the other. What does this tell us about the better ADC?

6. What does the *conversion time* figure for an ADC tell us about the device?

7. What other ADC specification does the conversion time affect?

8. What type of analog input signals require an ADC with a very high *sampling rate*?

9. What is the minimum sampling rate for an ADC with a 12kHz sinewave connected to its analog input?

10. What is the minimum sampling rate for an ADC with a complex signal on its analog input having sinewaves between 150Hz and 5600Hz?

11. Research question: Who is this minimum sampling theory named after?

12. Which of the four ADC structures mentioned in this workbook has the shortest conversion time?

13. Give two applications for the ADC structure that is the correct answer to the question above.

14. Which ADC structure is most widely used?

15. Which ADC structure is used for digital multimeters?

16. Why is the converter you named in the above question used for multimeters?

17. What is the term used to describe the difference between an ADC's sampled voltage and its closest centre voltage?

18. What problem does your answer to the previous question cause?

19. What does this problem sound like at the output of a DAC?

20. Calculate the resolution of a 12-bit ADC.

 Copyright: Not to be reproduced without permission

21. Define digital to analog conversion.

22. Draw the schematic symbol for a 6-bit DAC and label all inputs and outputs.

23. What does the *resolution* figure of a DAC tell us about the device?

24. What does the *settling time* figure of a DAC tell us about the device?

25. What does the *linearity* figure of a DAC tell us about the device?

26. What is the resolution of a 10-bit DAC?

Appendix 1

Flip flop circuits that implement counters

As already mentioned, counter circuits are implemented using flip flops. But, you're unlikely to see flip flop counter circuits because there are plenty of commercially available counter ICs. Despite this, we need to briefly examine the operation of flip flop counter circuits to understand the difference between asynchronous and synchronous counters. You'll not be asked to draw these discrete flip flop counter circuits in an exam nor will you be asked to explain their operation.

Asynchronous counters

Figure 1 below shows the circuit of a 2-bit (Mod-4) up-counter implemented using J-K flip flops (the flip flops are drawn back-to-front and that is acceptable practice). Notice that the *J* and *K* inputs of both flip flops are hard-wired to logic-1 and so they're both being operated in the toggle mode. (This means that D-type flip flops can be used instead of J-K flip flops as they can be wired to toggle too - see page 12-11 in the Part 1 workbook.)

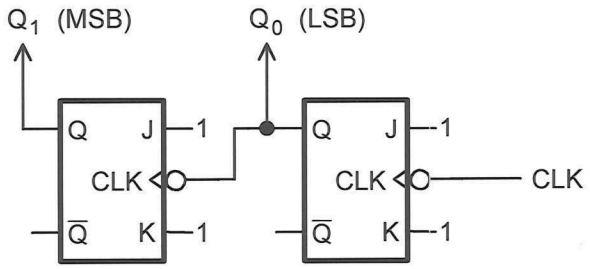


Figure 1 *A Mod-4 asynchronous up-counter implemented using J-K flip flops*

The operation of the circuit in Figure 1 is as follows. Assuming the outputs start at 00, the clock signal's first negative edge triggers the first flip flop and its output (Q_0) toggles to a logic-1. This makes the number on the outputs 01. The clock signal's second negative edge triggers the first flip flop again and so Q_0 toggles back to a logic-0. The act of Q_0 changing from a logic-1 to a logic-0 triggers the second flip flop and so its output toggles to logic-1. This makes the number on the outputs 10.

The clock signal's third negative edge triggers the first flip flop again and so Q_0 toggles to a logic-1 again which makes the number on the outputs 11. Finally, the clock signal's fourth negative edge triggers the first flip flop again and so Q_0 toggles back to a logic-0 which in turn triggers the second flip flop again and its output toggles back to logic-0 too. This makes the number on the outputs 00.

A Mod-8 counter can be made be simply connecting the clock of another toggling flip flop to the second flip flop's output. Connecting the clock of a fourth flip flop to the output of the third gives a Mod-16 counter and this process can be repeated indefinitely.

Figure 2 below shows the circuit of a Mod-16 asynchronous up-counter for your interest. Moduli greater than 16 (eg. 32, 64, 128 and so on) can be achieved by simply tacking on more flip flops in the same way.

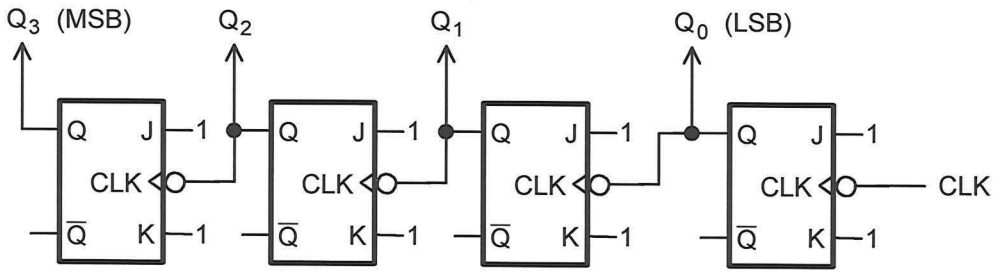


Figure 2 *A Mod-16 asynchronous up-counter implemented using J-K flip flops*

Why are these counter said to be "asynchronous"?
Regardless of the number of flip flops that make up the counter, the clock signal only ever triggers the first flip flop. The second flip flop is triggered by the first flip flop's Q-output, the third flip flop is triggered by the second flip flop's Q-output and so on in a domino-like effect.

Importantly, this means that the flip flops are never triggered at the same time so they are not synchronised to the clock or each other. For this reason, this type of counter is said to be asynchronous. As an aside, another name for the asynchronous counter is the *ripple* counter which is a reference to the way the clock signal ripples through the cascaded flip flops one after another.

Synchronous counters

Figure 3 below shows the circuit of a 2-bit (Mod-4) synchronous up-counter implemented using J-K flip flops.

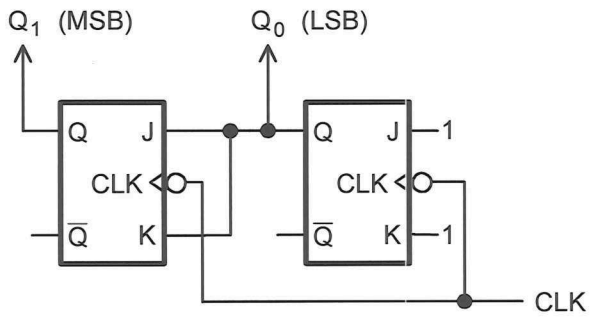


Figure 3 *A Mod-4 synchronous up-counter implemented using J-K flip flops*

Notice that the clock inputs of both flip flops are connected together and to the clock signal. This means that both flip flops will be triggered at exactly the same time and so they are synchronised to the clock and each other. For this reason, this type of counter is said to be synchronous.

Notice also that the first flip flop is wired to operate in the toggle mode but the second flip flop's mode of operation will depend on the first flip flop's output (Q_0). When Q_0 is a logic-0, the second flip flop's output wont change when it is clocked. When Q_0 is a logic-1, the second flip flop's output toggles when it is clocked.

The operation of the circuit in Figure 3 is as follows. Assuming the outputs start at 00, the clock signal's first negative edge triggers both flip flops and Q_0 toggles to a logic-1 but Q_1 doesn't change because, at the time of the clock, Q_0 is a logic-0. This makes the number on the outputs 01.

The clock signal's second negative edge triggers both flip flops and this time they both toggle. In the process, Q_0 goes back to a logic-0 and Q_1 goes to logic-1 making the number on the outputs 10.

The clock signal's third negative edge triggers both flip flops but this time only Q_0 toggles making the number on the outputs 11. Finally the clock signal's fourth negative edge triggers both flip flops and this time they both toggle. In the process, both Q_0 and Q_1 go to logic-1 making the number on the outputs 00.

Synchronous counters with a modulus greater than four (eg Mod-8, Mod-16, Mod-32 and so on) are also possible but designing the circuit involves a little more than simply tacking toggling flip flops onto the end of the circuit like asynchronous counters. Figure 4 below shows the circuit of a Mod-16 synchronous up-counter for your interest. As you can see, once the modulus of the circuit is greater than 4, the synchronous counter is much more complex than the asynchronous counter.

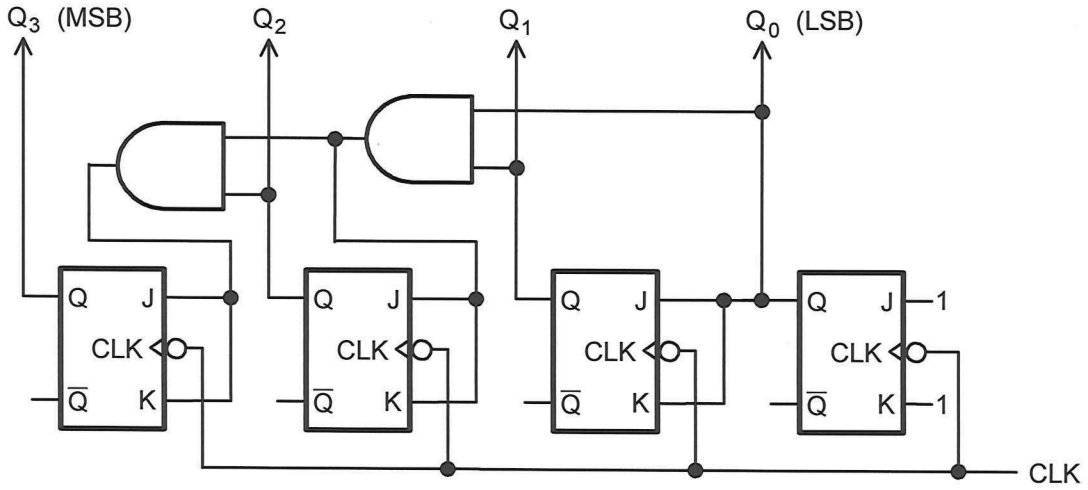


Figure 4 *A Mod-16 synchronous up-counter implemented using J-K flip flops and some extra necessary combinational logic*

The problem with asynchronous counters

As has just been discussed, the clock signal to the input of an asynchronous counter ripples through the circuit triggering the flip flops one after another. Importantly, this means that when the counter is clocked and the number on its output changes, the bits that make up the number change one at a time from the LSB to the MSB. This causes invalid numbers to very briefly punctuate the count sequence whenever two or more bits change logic level on a count.

For examples of this effect, consider the operation of a simple 2-bit (Mod-4) asynchronous up-counter. The circuit's count sequence is supposed to be 00, 01, 10 and 11 then back to 00 and so on. So, when the output of this counter is 01, the next clock pulse should produce the number 10. But, because the clock ripples through the flip flops, the count sequence actually goes from 01 to 00 for a very brief moment then to 10. Similarly when the counter's output is 11, the next clock pulse should produce the number 00 but the number 10 appears very briefly on the outputs first.

The existence of these invalid numbers can be demonstrated using a timing diagram for a 2-bit counter with the propagation delay for each output exaggerated (relative to the length of the clock) as shown in Figure 5 below.

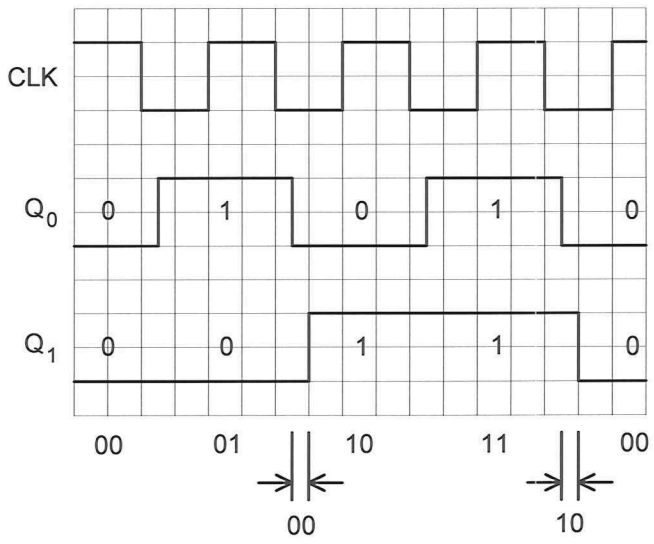


Figure 5 *The invalid numbers on the output of a 2-bit asynchronous (ripple) counter*

By exaggerating the propagation delay we can see that the numbers 00 and 10 exist (albeit briefly) in the count sequence where they shouldn't.

If the counter's output is driving a set of LEDs or a 7-segment display we wouldn't see these invalid numbers because they exist for such a short amount of time. They only last for the duration of a flip flop's propagation delay which is between 1.7ns and 70ns depending on the logic family/sub-family of the IC. As long as sufficient time is left for the outputs to *settle* before reading the number then, for most counter applications, these invalid numbers are not a problem. However, in some situations these invalid numbers can cause problems and disrupt the operation of whatever device the output is driving.

Synchronous counters don't experience this problem because all of the flip flops are connected to the one clock and so are all triggered at the same time causing their outputs to all change at the same time. For this reason, synchronous counters are often preferred to asynchronous counters.

Copyright: Not to be reproduced without permission

Appendix 2

Circuits that implement encoders and decoders

Most of the encoders and decoders that you'll encounter as a technician will be inside an IC. However, discrete encoders and decoders are shown in this appendix for your interest. You'll not be asked to recognise, draw or explain these circuits for the tests in this module.

Encoders

Figure 1 below shows the circuit that implements a 4-line-to-2-line encoder with active high inputs and outputs.

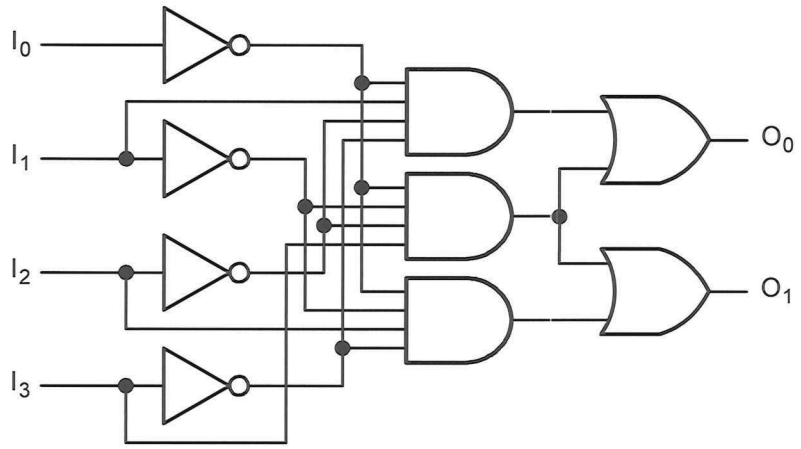


Figure 1 *A simple 4-line-to-2-line encoder*

To verify that this circuit is an encoder you could develop a truth table for it by writing the codes 0000, 0001, 0010, 0100 and 1000 on the circuit's input one code at a time and chasing the logic levels through to the outputs to see which binary numbers the code generates. If you do, you find that the circuit operates in the same way as the device in Figure 1 on page 4-2 (it's operation is summarised on page 4-3).

However, there's a better way of developing the truth table for this circuit and you learnt about this method earlier. Instead of chasing logic levels through the circuit it is faster and simpler to chase the Boolean expressions through the circuit deriving expressions for the outputs then using them to write the circuit's truth table. This process is shown on the next page.

First, derive the Boolean expressions for the outputs.

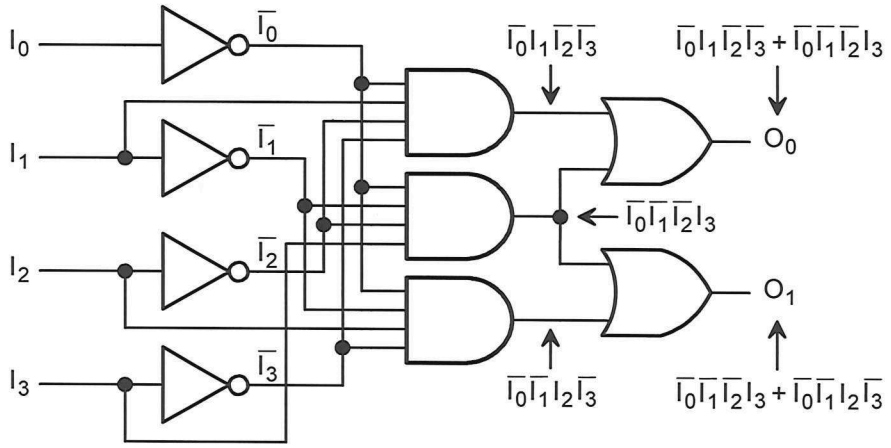


Figure 2 *The circuit in Figure 1 with Boolean expressions shown for each node*

Then, plot the truth table for the each of the encoder's outputs using the corresponding expressions. This gives Tables 1 and 2 below.

Table 1

I_0	I_1	I_2	I_3	O_0	
1	0	0	0	0	
0	1	0	0	1	$\leftarrow O_0 = \overline{I_0} I_1 \overline{I_2} \overline{I_3}$
0	0	1	1	0	
0	0	0	1	1	$\leftarrow O_0 = \overline{I_0} \overline{I_1} \overline{I_2} I_3$

Table 2

I_0	I_1	I_2	I_3	O_1	
1	0	0	0	0	
0	1	0	0	0	
0	0	1	1	1	$\leftarrow O_1 = \overline{I_0} \overline{I_1} I_2 \overline{I_3}$
0	0	0	1	1	$\leftarrow O_1 = \overline{I_0} \overline{I_1} \overline{I_2} I_3$

Finally, combine these tables into one truth table. This gives Table 3 below and, as you can see, the device is a 4-line-to-2-line encoder (though not a priority encoder).

Table 3

Inputs				Outputs	
I_0	I_1	I_2	I_3	O_1	O_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

Decoders

Figure 3 below shows the circuit that implements a 1-of-4 decoder with active high outputs.

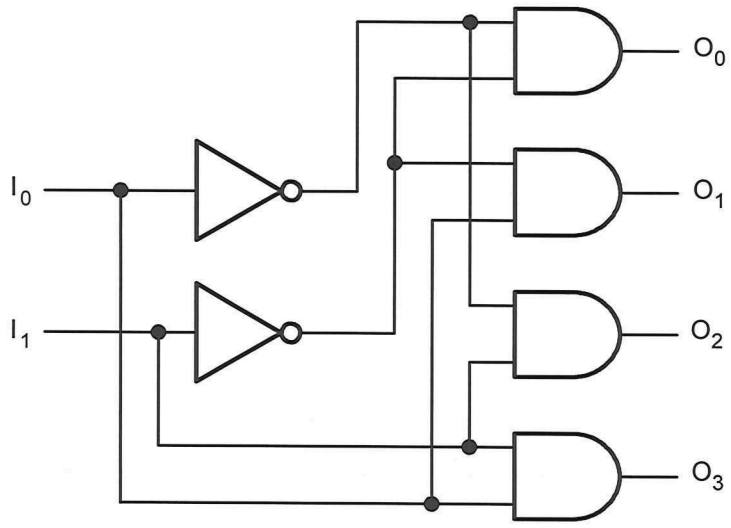


Figure 3 *A 1-of-4 decoder*

Again, it's probably better to analyse the circuit using Boolean algebra and Figure 4 below shows the Boolean expressions for each node in the circuit.

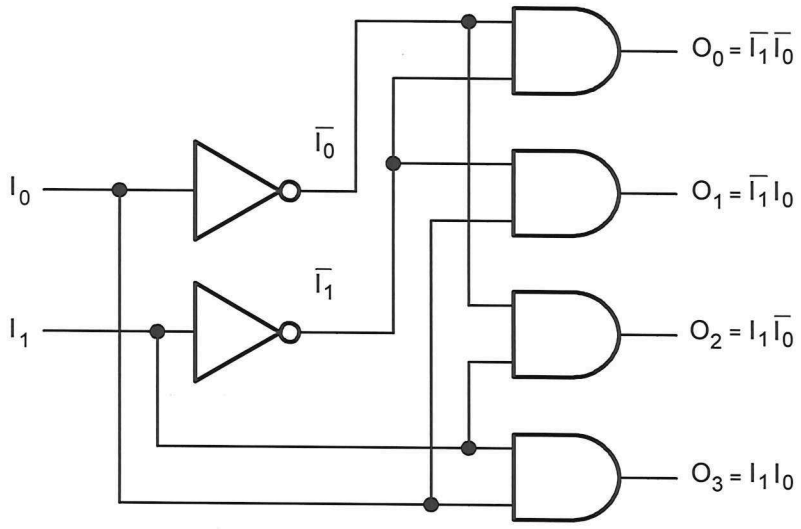


Figure 4 *The circuit in Figure 3 with Boolean expressions shown for each node*

Then, plot the truth table for the each of the decoder's outputs using the corresponding expressions. This gives Tables 4 to 7 below.

Table 4

I_1	I_0	O_0
0	0	1
0	1	0
1	0	0
1	1	0

$O_0 = \overline{I_1}\,\overline{I_0}$

Table 5

I_1	I_0	O_1
0	0	0
0	1	1
1	0	0
1	1	0

$O_0 = \overline{I_1}\,I_0$

Table 6

I_1	I_0	O_2
0	0	0
0	1	0
1	0	1
1	1	0

$O_0 = I_1\,\overline{I_0}$

Table 7

I_1	I_0	O_3
0	0	0
0	1	0
1	0	0
1	1	1

$O_0 = I_1\,I_0$

Finally, combine these tables into one truth table. This gives Table 8 below and, as you can see, the device is a 1-of-4 decoder.

Table 8

Inputs		Outputs			
I_1	I_0	O_0	O_1	O_2	O_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Appendix 3

Circuits that implement multiplexers

Most, if not all, of the multiplexers and demultiplexers that you'll encounter as a technician will be inside an IC. However, a discrete gate multiplexer circuit is shown in this appendix for your interest. You'll not be asked to draw, recognise or explain this circuit for the tests in this module.

A demultiplexer circuit is not shown here because, as the notes in Section 5 explain, decoder circuits are used to implement demultiplexers and an example of a discrete gate decoder is shown in Appendix 2.

Multiplexers

Figure 1 below shows the circuit that implements a 2-input demultiplexer.

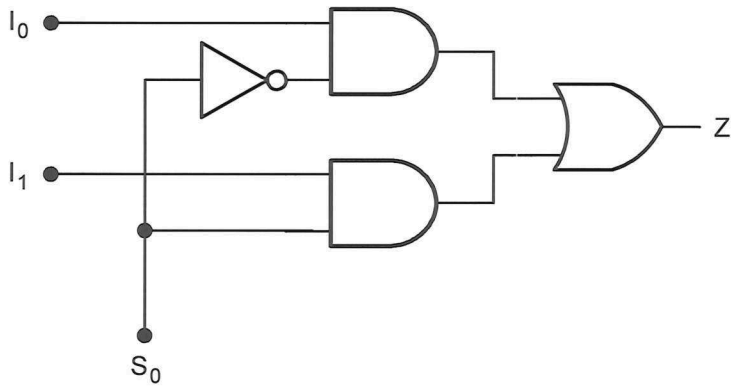


Figure 1 *A 2-input demultiplexer*

To verify that this circuit is a demultiplexer you could develop a truth table for it by writing logic-0 on the select input (S_0) then writing the codes 00, 01, 10 and 11 on the other inputs one code at a time and chasing the logic levels through to the outputs to see what happens on Z. Then, you'd have to write logic-1 on the select input followed by 00, 01, 10 and 11 on the other inputs to see what happens on Z. If you do, you find that the circuit operates in the same way as the device in Figure 2 on page 5-4 (it's operation is summared on page 5-4 also).

However, it is probably easier and quicker to analyse this circuit by chasing the Boolean expressions through the circuit deriving expressions for the outputs then using them to write the truth table. This process is shown on the next page.

First, derive the Boolean expressions for the outputs.

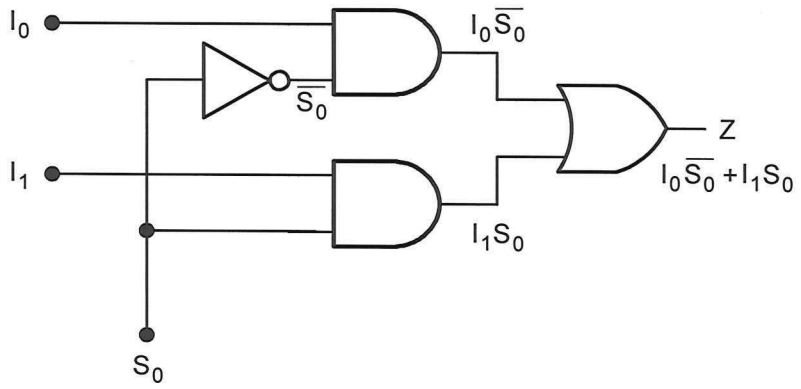


Figure 2 *The circuit in Figure 1 with Boolean expressions shown for each node*

Then, plot the truth table for the output using its Boolean expression. This gives Table 1 below.

Table 1

Inputs			Output	
S_0	I_1	I_0	Z	
0	0	0	0	
0	0	1	1	← $Z = I_0\overline{S_0}$
0	1	0	0	
0	1	1	1	← $Z = I_0\overline{S_0}$
1	0	0	0	
1	0	1	0	
1	1	0	1	← $Z = I_0S_0$
1	1	1	1	← $Z = I_0S_0$

If you look closely at Table 1 you'll notice that when S_0 is a logic-0, Z is always the same as I_0 and when S_0 is a logic-1, Z is always the same as I_1. That being the case, this table can be simplified as shown in Table 2 below and, as you can see, the device is a 2-input demux.

Table 2

Inputs			Output
S_0	I_1	I_0	Z
0	×	0	0
0	×	1	1
1	0	×	0
1	1	×	1

Appendix 4

ADC structures

There are several ADC structures including:

- Digital ramp converter
- Successive approximation converter
- Dual slope converter
- Flash converter

The following notes explain their operation.

The Digital Ramp Converter

The basic block diagram of a Digital Ramp ADC is shown in Figure 1.

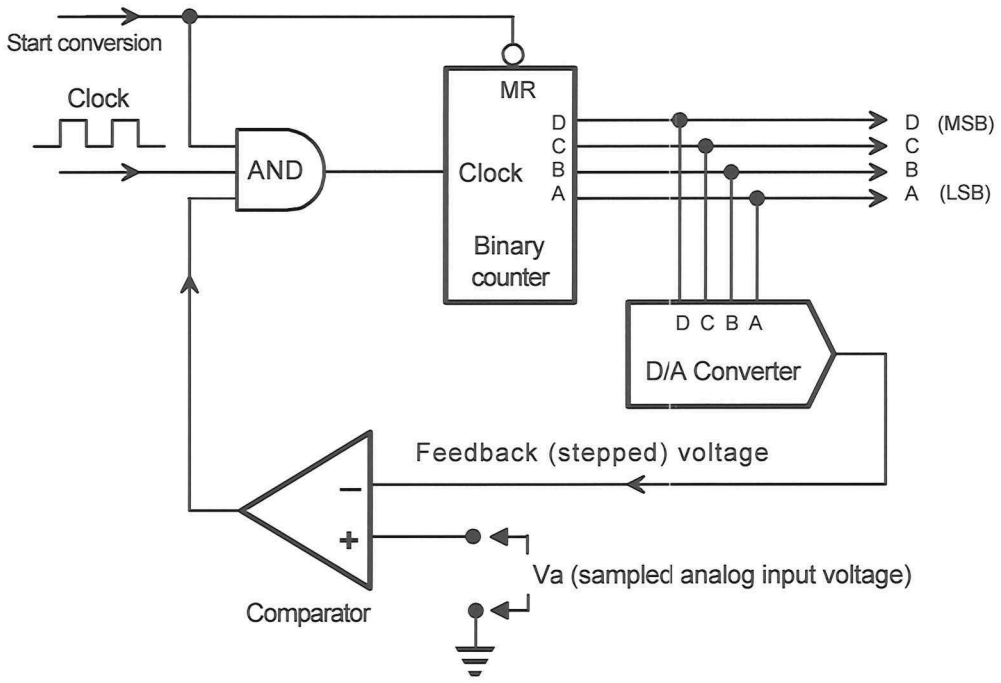


Figure 1 *The basic block diagram of a digital ramp ADC*

Operation

- When the *Start Conversion* input is asserted with a logic 0, the binary counter is reset and the output of the AND gate is forced low. This isolates the counter's clock input from the clock signal.

- With the binary counter reset to 0000, the output of the DAC and the comparator's inverting input will be 0V. As the analog input voltage is connected to the comparator's non-inverting input the voltage on inverting input is smaller so and the comparator's output is a logic 1.

- The DAC is now ready to start a new conversion and these conditions are shown in Figure 2 below.

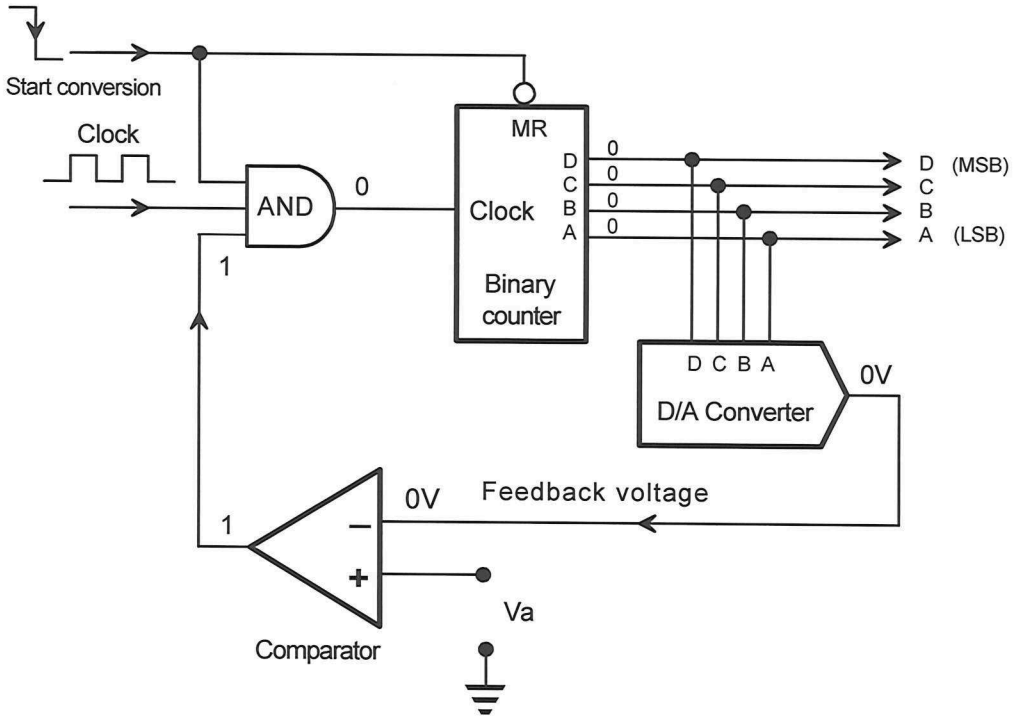


Figure 2 *The digital ramp DAC is readied for the start of a new conversion*

- When the *Start Conversion* line returns to a logic 1, two of the three AND gate's inputs are now logic 1.

- Under these conditions, when the clock signal is a logic 0 the output of the AND is a logic 0. When the clock signal is a logic 1 the output of the AND gate is a logic 1. This means that the clock signal is reproduced at the output of the AND gate (see Figure 3 on the next page).

- The counter (which was reset to 0000) now counts up by one each time the output of the AND gate changes from a logic 0 to a logic 1.

- Every new number at the counter's output is converted to an analog voltage by the DAC whose output resembles a set of stairs or a "digital ramp" (see Figure 3).

- The output of the DAC is continuously compared with the analogue input voltage (*Va*).

- While the output of the DAC is lower than *Va* the output of the comparator remains logic 1 and the counter is allowed to continue counting.

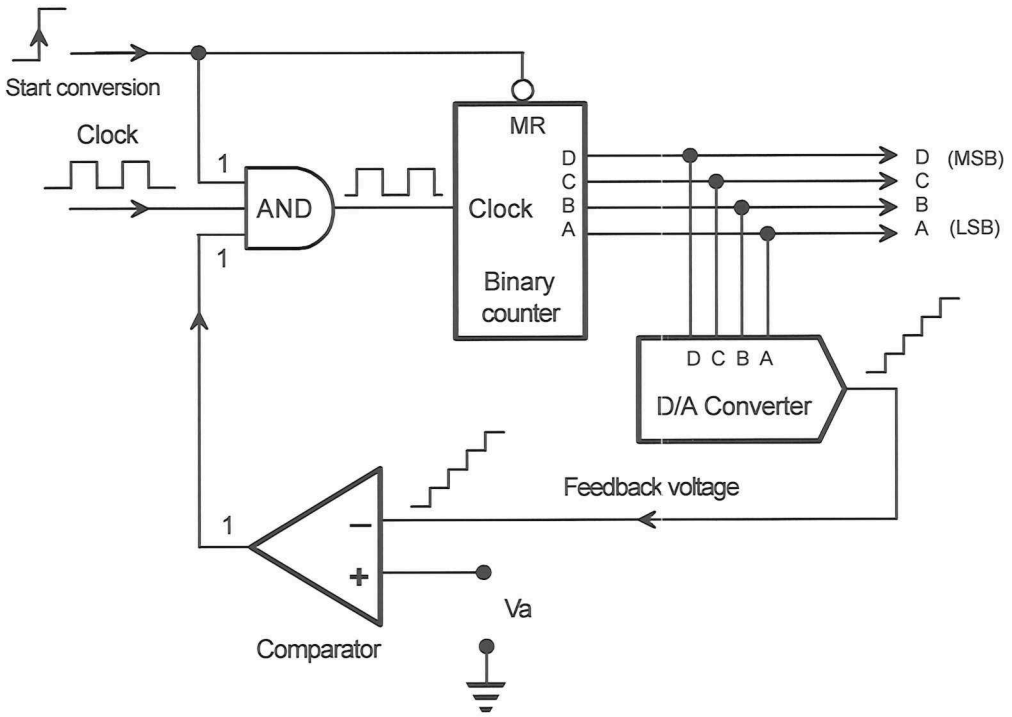


Figure 3 *The conditions around the circuit during the conversion process*

- The moment the output of the DAC is larger than *Va* the output of the comparator goes to logic 0. This forces the output of the AND gate to a logic 0 stopping the clock signal from getting through to the clock input on the counter.

- Without a clock signal the counter stops counting and the binary code at its output is the digital representation of the analog input voltage.

- To start another conversion the *Start Conversion* line needs to be taken low momentarily. This resets the counter and repeats the cycle just described.

This process can be summarised in terms of what goes on around the circuit in relation to the output of the DAC and is shown in Figure 4 below.

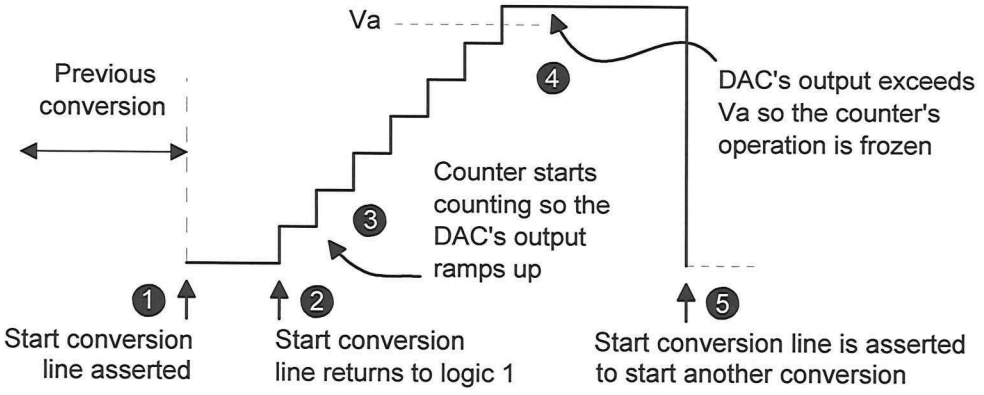


Figure 4 *The digital ramp that appears at the output of the DAC during the conversion process*

Recall that, by definition, the digital output of an ADC must be proportional to *Va* and this applies here. If *Va* is small, it only takes a few clock pulses before the output of the counter produces a voltage at the output of the DAC that exceeds *Va*. As a result, the digital code is a small number. However, if *Va* is large, it takes a lot of clock pulses before the output of the counter produces a voltage at the output of the DAC that exceeds *Va*. The resulting digital code is a large number.

The operation of the digital ramp ADC is analogous to the operation of a toilet cistern. The start conversion pulse is like pushing the flush button. The counter resetting to zero is like the cistern emptying into the bowl. The incrementing counter is like the cistern filling up, and the stopping of the counter is like the stopping of the water flowing into the cistern when it's full.

That said, there is a one important difference between the operation of the digital ramp ADC and a toilet cistern. A toilet cistern fills up to the same point every time whereas the point at which the counter stops varies (and depends on the size of the analog input voltage).

There are several problems with the digital ramp ADC. First, the conversion time is variable and depends on the size of *Va*. To explain, if *Va* is small then it only takes a few clock pulses for the conversion process to be completed. As *Va* gets bigger it takes more clock pulses to complete the conversion and hence it takes more time.

This leads to a second problem: if *Va* is large then the conversion time is relatively slow. A third problem occurs if there is any noise superimposed on the analog input signal (which invariably there is!). Noise causes *Va* to swing above and below its actual voltage. The noise may make Va seem either smaller or larger than it actually is for an instant which may lead to conversion to the wrong number (which could be out by ±1).

The Successive Approximation Converter

The basic block diagram of a Successive Approximation ADC is shown in Figure 5.

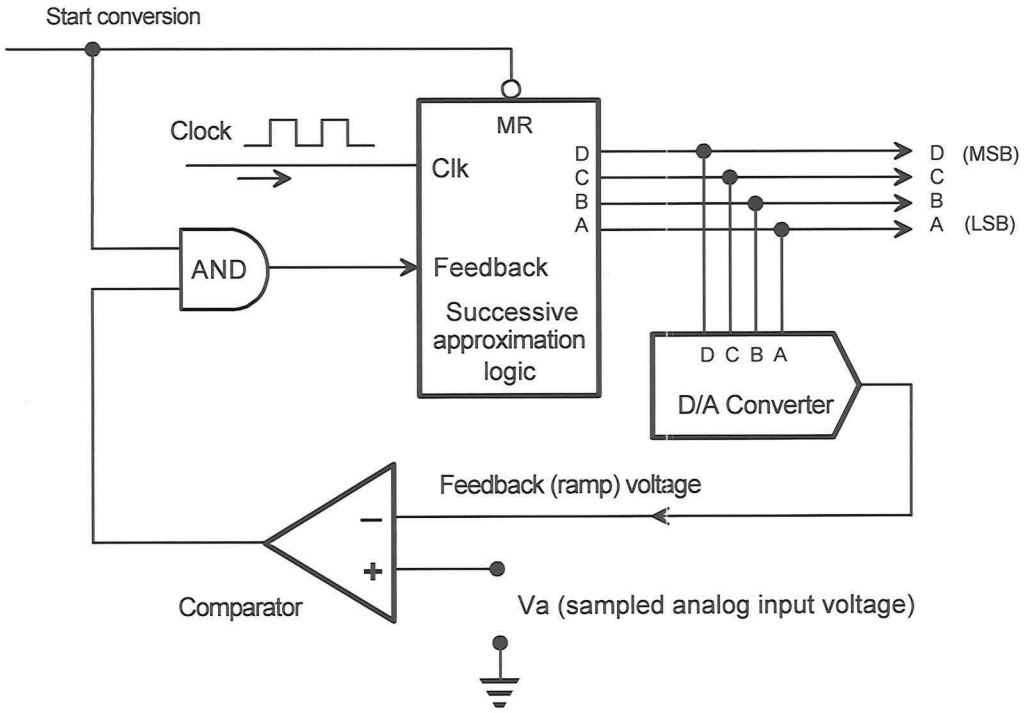


Figure 5 *The basic block diagram of a successive approximation ADC*

The operation of the successive approximation converter (SAC) is very similar to the digital ramp converter. The difference between them is in how the binary code is generated. In the digital ramp converter a counter is used whereas a successive approximation logic circuit is used for the SAC.

Operation

- When the *Start Conversion* line is asserted with a logic 0, the successive approximation logic is reset. The 0000 at its output produces 0V at the output of the DAC. This is less than *Va* and hence the output of the comparator is a logic 1.

- When the *Start Conversion* line is returned to a logic 1, the output of the AND gate and hence the feedback input on the successive approximation logic becomes a logic 1.

- On the first clock pulse after the *Start Conversion* line is returned to a logic 1 the successive approximation logic asserts the most significant bit (that is, makes it a logic 1).

- At this point one of two things can happen:
  - If asserting the MSB produces an output from the DAC that is smaller than *Va*, the comparator's output and hence the feedback input remains a logic 1. The successive approximation logic detects this and keeps the MSB asserted.
  - If asserting the MSB produces an output from the DAC that is bigger than *Va*, the comparator's output and hence the feedback input returns to a logic 0. The successive approximation logic detects this and resets the MSB (that is, makes it a logic 0 again).
- On the next clock pulse the successive approximation logic asserts the next most significant bit. Again, the bit is kept a logic 1 or reset depending on whether the code produces an output from the DAC that is smaller or bigger than *Va*.
- This procedure is followed for each successive bit up to and including the LSB.
- Once all of the bits have been "tested" in this way, the code at the output is a digital code that is proportional to *Va*.

The operation of the successive approximation ADC may be better understood with an example. Suppose the step voltage of the DAC in the circuit of Figure 5 is 500mV. The tables below show how each bit - starting with the MSB - is asserted and is either kept a logic 1 or returned to a logic 0 depending on whether the resulting change in the output of the DAC is smaller or larger than *Va*. Table 1 shows the sequence of events for a *Va* of 2.6V and Table 2 for a *Va* of 5.2V.

Table 1

Code	Output of the DAC	Is the bit reset?
1000	4V	Yes
0100	2V	No
0110	3V	Yes
0101	2.5V	No
0101	2.5V	Finished

Clock pulse
t + *1*
t + *2*
t + *3*
t + *4*
t + *5*

Table 2

Code	Output of the DAC	Is the bit reset?
1000	4V	No
1100	6V	Yes
1010	5V	No
1011	5.5V	Yes
1010	5V	Finished

From these examples it can be seen that the digital output is proportional to *Va*. A *Va* of 2.6V produces the number 0101 (or 5_{10}) and a *Va* of 5.2V (which is twice as big as 2.6V) produces the number 1010 (or 10_{10}).

If the successive approximation method seems a little odd, you might be surprised to know that you use the same method when giving somebody an exact amount of money. For example, if you wanted to give a friend $5.66 you unconsciously ask yourself the following questions: Do I give him/her a $10 bill? No, it's too much. Do I give him/her a $5 bill? Yes. Do I give him/her a $2 coin? No, it's too much. Do I give him/her a $1 coin? No, it's too much. Do I give him/her a 50c coin? Yes. Do I give him/her a 20c coin? No, too much . . . and so on. This is successive approximation.

The successive approximation ADC has two significant advantages over the digital ramp ADC (and the dual slope ADC). First, the conversion time is independent of *Va*. That is, it takes the same amount of time to convert a small *Va* as a large *Va*. The conversion time is always equal to *n+1* where n is the number of inputs. The other advantage is the conversion time is fairly fast (typically 100µs).

The complexity of the successive approximation logic is said to be a disadvantage of the successive approximation converter but this is not really a concern these days given ADCs are cheaply available in integrated circuits. In fact, the successive approximation ADC is the most commonly used ADC inside ICs these days. However, the successive approximation ADC suffers the same problems with noise as the digital ramp ADC.

The Dual Slope Converter

The basic block diagram of a Dual Slope ADC is shown in Figure 6 below.

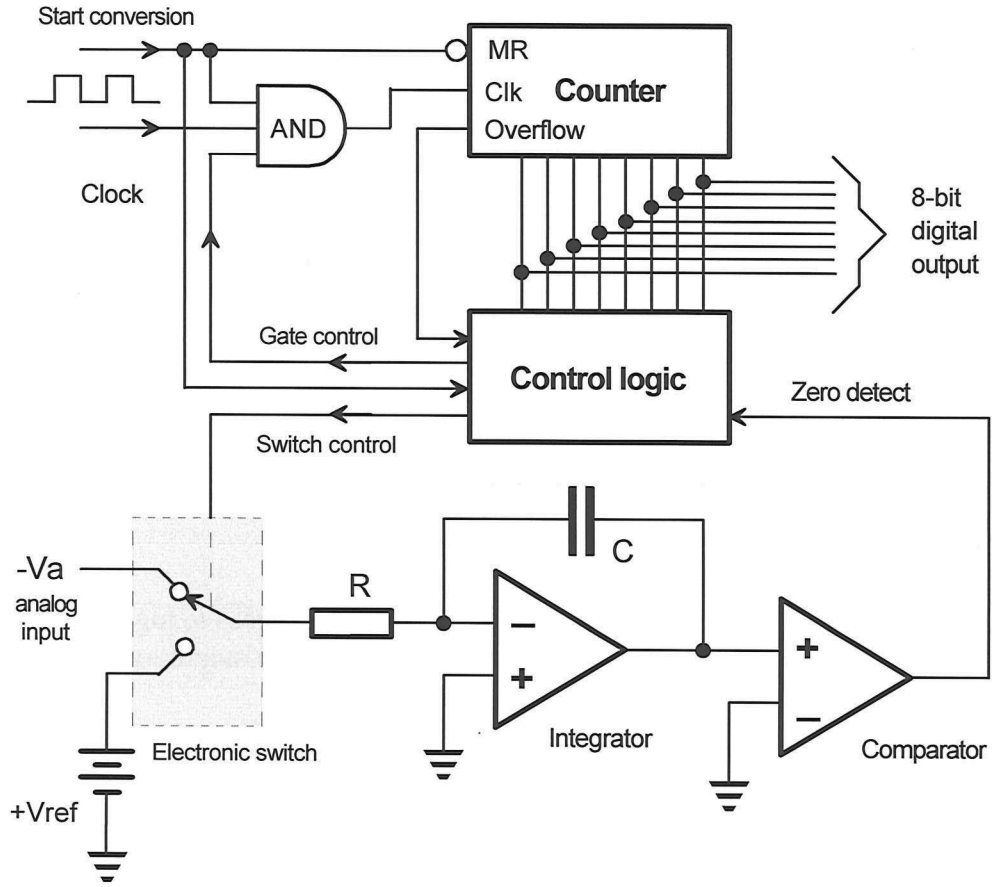


Figure 6 *The basic block diagram of the dual slope ADC*

Operation

- At the completion of the previous conversion, the electronic switch was connected to the reference voltage, the output of the integrator was fractionally below 0V and the output of the comparator was logic 0. The control logic had detected this and asserted it's output to the AND gate to logic 1 also.

- When the *Start Conversion* line is asserted, the counter is reset and the output of the AND gate is forced low isolating the clock signal from the clock input to the counter.

- When the *Start Conversion* line is returned to a logic 1, the control logic detects this and switches the electronic switch so that the input to the integrator is connected to *Va*. At the same time, two of the three AND gate's inputs are logic 1.

- Under these conditions, when the clock signal is a logic 0 the output of the AND is a logic 0. When the clock signal is a logic 1 the output of the AND gate is a logic 1. This means that the clock signal is reproduced at the output of the AND gate.

- The counter now counts up by one on each negative transition at the output of the AND gate.

- At the same time, the negative voltage at *Va* causes a positive going ramp at the output of the integrator (see Figure 7 below) which makes the output of the comparator a logic 1.

- The counter is allowed to count up until it "overflows" (that is, until it reaches 1111-1111 then goes to 0000-0000).

- When this happens, an overflow (or carry-out) signal is sent to the control logic. In response, the control logic switches the electronic switch from *Va* to the reference voltage.

- As V_{REF} is a positive voltage the output of the integrator produces a negative going ramp (see Figure 7).

- When the ramp goes below 0V the output of the comparator goes to logic 0 which the control logic detects and outputs a logic 0 to the AND gate forcing its output low and stopping the clocking action of the counter.

- When the counter stops counting, the code at its output is a binary code that is proportional to *Va*.

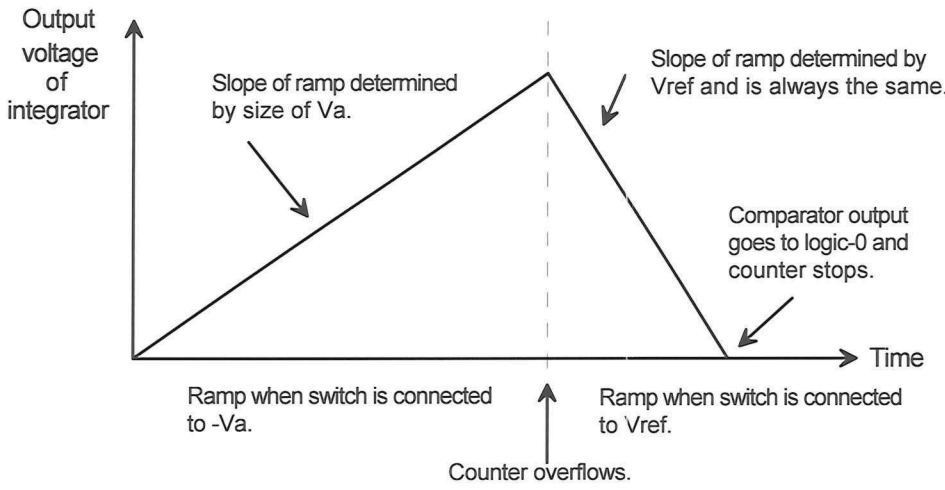

Output voltage of integrator
Slope of ramp determined by size of Va.
Slope of ramp determined by Vref and is always the same.
Comparator output goes to logic-0 and counter stops.
Time
Ramp when switch is connected to -Va.
Ramp when switch is connected to Vref.
Counter overflows.

Figure 7 *Timing of events around a dual slope ADC in relation to the output of the integrator*

Figure 8 below what happens at the output of the integrator for a small and large value of *Va*. If *Va* is small then, when the counter overflows, the voltage at the output of the integrator is relatively small and so it wont take long for it to return to 0V and stop the counter. Consequently, the counter will not have had long to count up and so the number on its output is small.

If *Va* is large then, when the counter overflows, the voltage at the output of the integrator is relatively large so it takes longer for it to return to 0V and stop the counter. Consequently the counter counts for longer and has a bigger number on its output. In this way the digital code in the dual slope ADC is proportional to the analog input voltage.

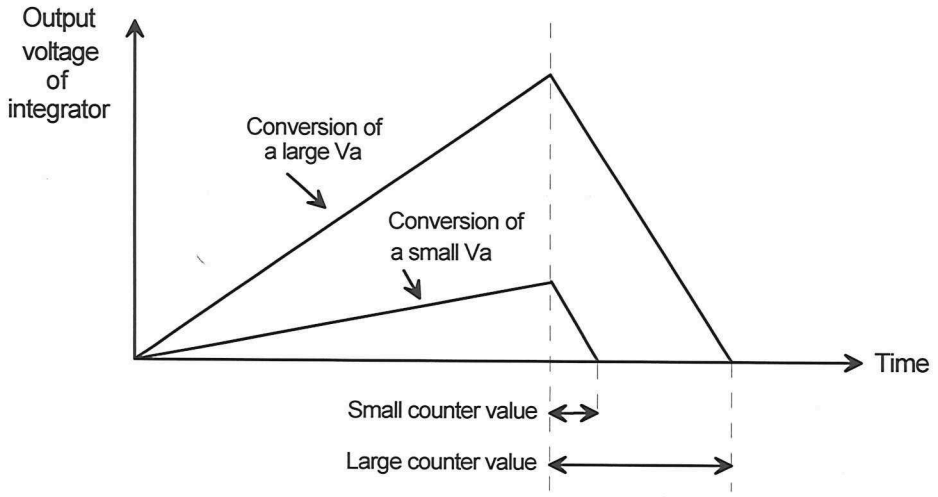


Figure 8 *The output of the integrator during a single conversion for small and large values of Va*

The dual slope ADC has a significant advantage over the digital ramp and successive approximation (and Flash) ADCs; it has an excellent noise immunity. Any noise on *Va* (which mostly swings above and below the actual *Va* signal by an equal amount) is averaged out by the integrator so it doesn't affect the conversion process. This makes the dual slope ADC fairly precise because each particular input voltage will produce the same digital code each time it is converted.

The "averaging" ability of the integrator also gives the circuit an excellent immunity to temperature variations and drift in component values due to age. For these reasons, the dual slope ADC is ideal for making measurements and is used almost exclusively in digital multimeters (the chips regularly used in DMMs are the 7106 and the 7116).

The disadvantages of the dual slope ADC include a variable conversion time that is also usually slow (typically between 10ms and 100ms).

The Flash Converter

The basic block diagram of a Flash ADC is shown in Figure 9 below.

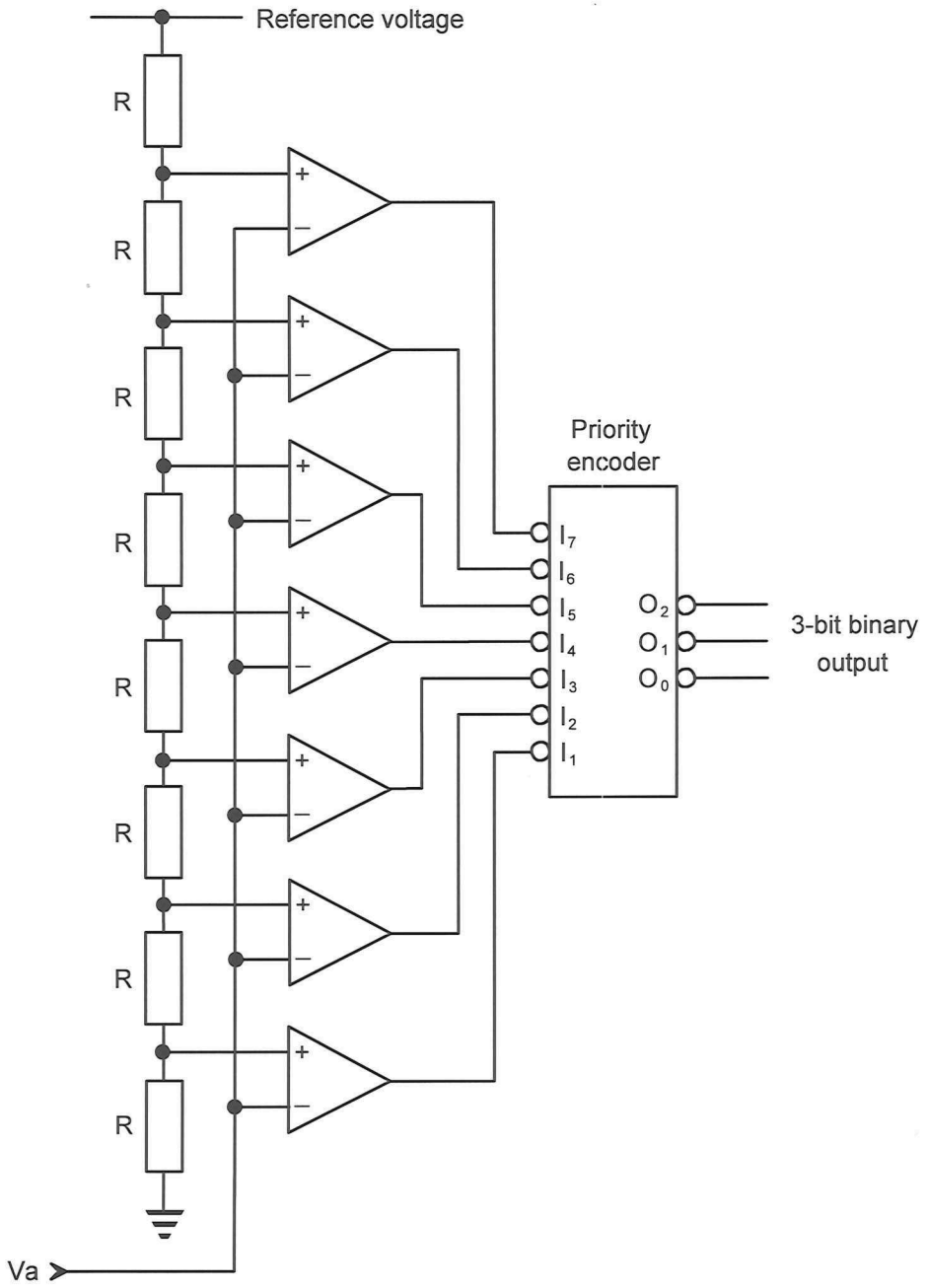


Figure 9 *The basic block diagram of a Flash ADC*

Operation

- A circuit reference voltage is divided by a voltage divider network into individual op amp reference voltages. All of the resistors are the same value so the potential difference across each resistor is the same. This means that the voltages on the non-inverting inputs to the op amps (with respect to earth) increase in even voltages up to the reference voltage.

- The job of the comparators is to compare their individual op amp reference voltages with the analog input voltage.

- If an analog input (*Va*) is 0V, the voltage on the inverting input on all of the op amps is 0V. This means that all the inverting inputs to all of the op amps is a voltage less than the non-inverting inputs so their outputs are all logic 1.

- With a logic 1 on the output of all the op amps, none of the inputs to the priority encoder are asserted. Consequently, the binary code on the encoder's output is 000.

- For analog input voltages greater than 0V, the outputs of all the op amps with a non-inverting input voltage less than *Va* go to logic 0.

- The priority encoder then produces a binary number whose value is equal to the number of the highest weighted input that is asserted.

The operation of the circuit may be better understood by doing an example with values. Figure 10 shows the Flash converter with a circuit reference voltage of 8V. This produces the individual op amp references shown. If *Va* is 4.5V the outputs of the first four op amps are logic 0. Input 4 of the priority encoder is the highest weighted input that is asserted so it outputs the binary number 100 (decimal 4).

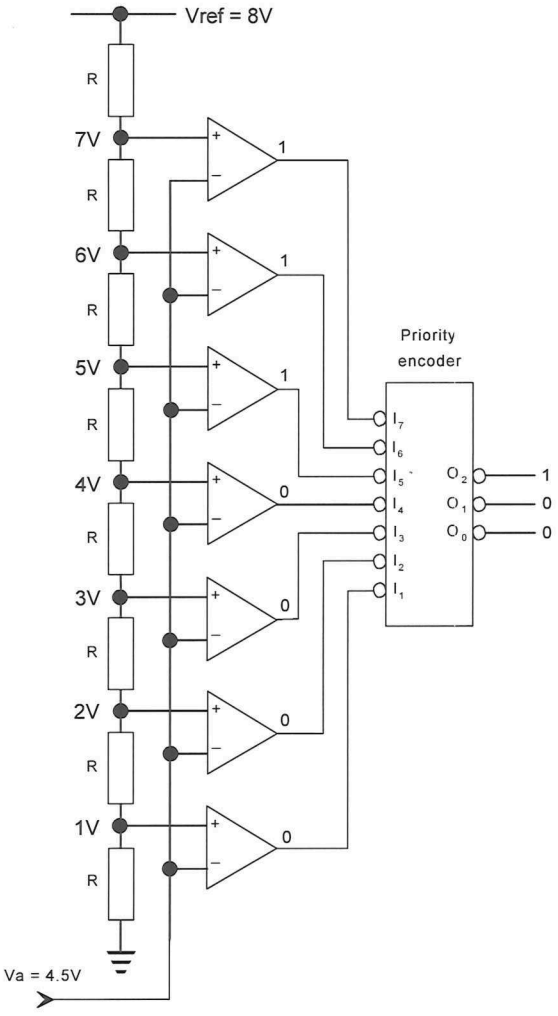


Figure 10 *A Flash ADC with an 8V reference and 4.5V analog input voltage*

The bigger the value of *Va*, the more op amp reference voltages it exceeds and the bigger the binary number generated. In this way the circuit produces a digital code that is proportional to the size of the analog input voltage.

The main advantage of the flash converter is that it is provides a very short conversion time compared to other ADCs. Typical conversion times for a fully integrated flash converter is 150ns.

The large number of comparators required for the circuit (2^n - 1 where n is the number of inputs to the ADC) is said to be a disadvantage of this ADC but this is not really a concern these days given ADCs are readily available in integrated circuits. However, the circuit has the same noise problems as both the digital ramp ADC and the successive approximation ADC.

Copyright: Not to be reproduced without permission

Appendix 5

DAC types

There are two basic DAC circuits:

- Summing amp
- R/2R ladder network

Summing amp DAC

As its name implies, the summing amp DAC relies on the operation of the *summing amplifier* which you learnt about in an earlier subject. Figure 1 below shows the circuit of a 2-input (or 2-bit) summing amp DAC.

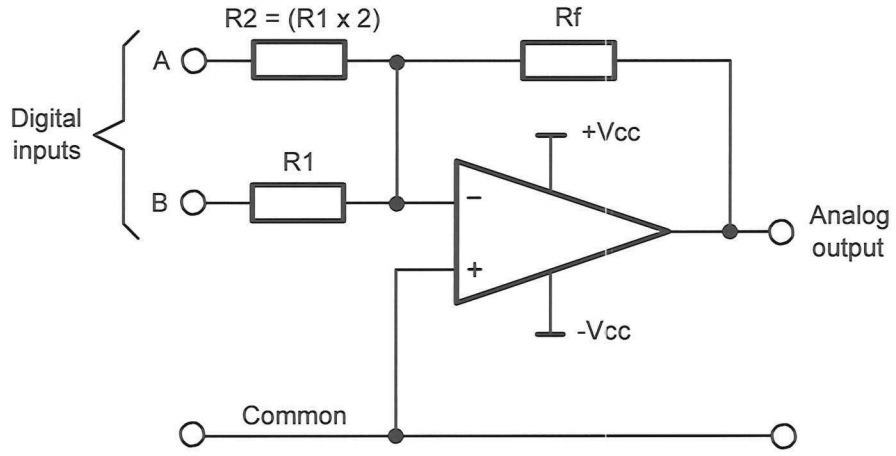


Figure 1 *A 2-bit summing amp digital to analog converter*

Notice that, although component values haven't been shown, the value of the input resistors relative to each other has been. That is, *R2* must be twice the value of *R1*. This relationship is crucial as you will see.

To explain how a summing amplifier can operate as a DAC, let's consider the 2-bit DAC circuit in Figure 2 below. It has values for resistors and switches that allow the inputs to be connected to either 5V (or logic-1) and 0V (or logic-0).

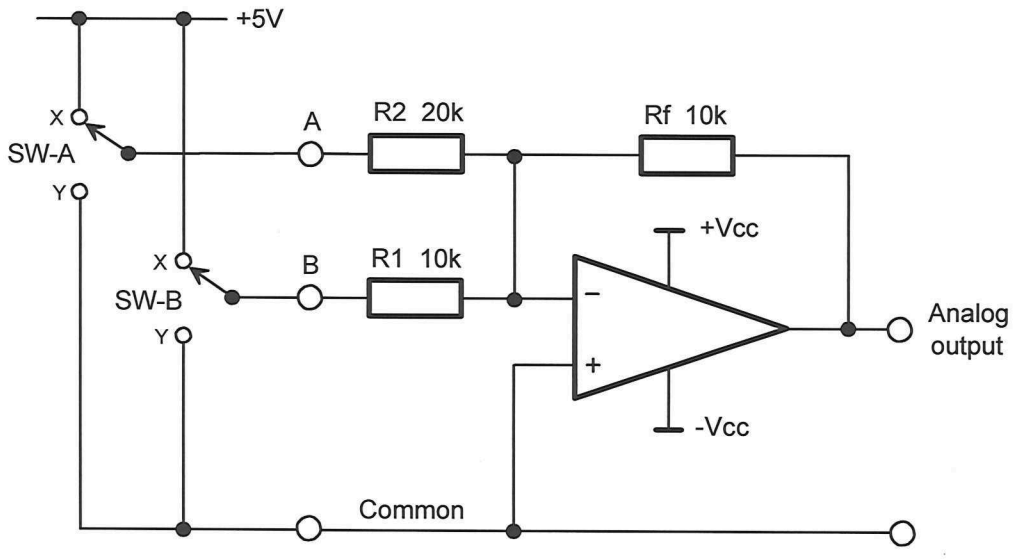


Figure 2 *A 2-bit summing amp DAC with switches on the inputs*

Recall that the output voltage of a summing amplifier can be calculated using the equation:

$$Vout = \left[Vin \times -\left(\frac{Rf}{R1}\right) \right] + \left[Vin \times -\left(\frac{Rf}{R2}\right) \right]$$

When both switches are in the "Y" position. The voltage on both of the inputs is 0V. So, *Vout* is:

$$Vout = \left[0V \times -\left(\frac{10k}{10k}\right) \right] + \left[0V \times -\left(\frac{10k}{20k}\right) \right]$$

$$Vout = [0V \times -1] + [0V \times -0.5]$$

$$Vout = 0V + 0V$$

$$Vout = 0V$$

When switch *A* is in the "X" position and switch *B* is in the "Y" position the *A* input has 5V connected to it and the *B* input has 0V connected to it. So, *Vout* is:

$$Vout = \left[0V \times -\left(\frac{10k}{10k}\right) \right] + \left[5V \times -\left(\frac{10k}{20k}\right) \right]$$

$$Vout = [0V \times -1] + [5V \times -0.5]$$

$$Vout = 0V + -2.5V$$

$$Vout = -2.5V$$

When switch *A* is in the "Y" position and switch *B* is in the "X" position the *A* input has 0V connected to it but the *B* input has 5V connected to it. So, *Vout* is:

$$Vout = \left[5V \times -\left(\frac{10k}{10k}\right) \right] + \left[0V \times -\left(\frac{10k}{20k}\right) \right]$$

$$Vout = [5V \times -1] + [0V \times -0.5]$$

$$Vout = -5V + 0V$$

$$Vout = -5V$$

Finally, when both switches are in the "X" position both inputs are connected to 5V. So, *Vout* is:

$$Vout = \left[5V \times -\left(\frac{10k}{10k}\right) \right] + \left[5V \times -\left(\frac{10k}{20k}\right) \right]$$

$$Vout = [5V \times -1] + [5V \times -0.5]$$

$$Vout = -5V + -2.5V$$

$$Vout = -7.5V$$

In summary, the four possible combinations of switch positions produce the following output voltages:

B	A	*Vout*
0V	0V	0V
0V	5V	-2.5V
5V	0V	-5V
5V	5V	-7.5V

If we substitute logic levels for voltages on the inputs the table looks like:

B	A	*Vout*
0	0	0V
0	1	-2.5V
1	0	-5V
1	1	-7.5V

This table demonstrates that the circuit very neatly meets the definition of a DAC. A binary number on the DAC's input has been converted to a voltage that is proportional to the number's value. That is, 00 produces the smallest output voltage, 01 produces the next smallest output voltage, 10 produces twice the output voltage produced by 01 and 11 produces the three times the output voltage produced by 01. The number 11 also produces the biggest (or *full-scale*) output voltage.

As mentioned in Section 6, the more bits on an ADC's output, the better the resolution (and the lower the quantisation noise). If a 4-bit ADC is used, a 4-bit DAC must also be used to convert the data back to analog. Figure 3 belows the summing amp wired to implement a 4-bit DAC.

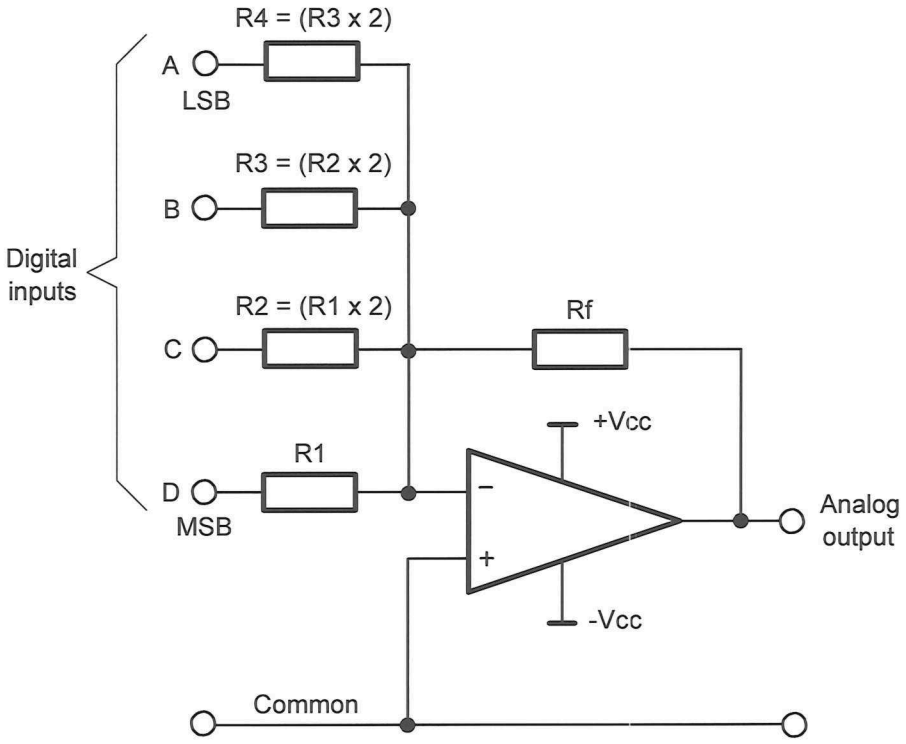
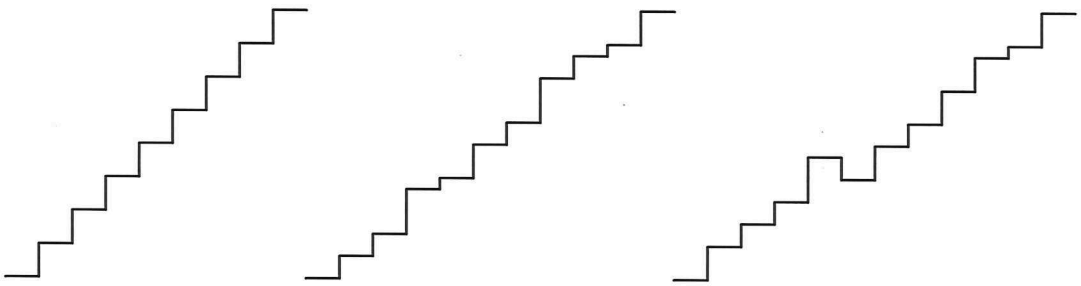


Figure 3 *A 4-bit summing amp DAC*

Notice that increasing the number of inputs bits to the summing amp DAC is a simple matter of increasing the number of input resistors. Importantly, each new additional resistor must have a value twice the size of the preceeding resistor. This makes sense when you think about it - each new column in a binary number has twice the weighting of the preceeding column.

Limitations of the summing amp DAC

There are several problems with the summing amplifier DAC that limits its practical use. First, if the circuit is being made from discrete components (that is, it's not in an IC) it may not be possible to get exactly the right value resistor for each input due to the E12 system of preferred resistor values. That is, although 2k2Ω is a preferred resistor value, 4k4Ω is not. The effect of using the nearest preferred resistor values would be to make the step voltages slightly different to each other as shown in Figure 4b below.



(a) Even step size (b) Uneven step size (c) Uneven step-size with reverse step!

Figure 4

The second limitation of the summing amp DAC exists because the input impedance of an inverting amplifier (from which the summing amp is derived) is equal to the value of the resistor on the input. That being the case, the input resistance of each input to a summing amp DAC is different because each input's resistor is a different value. As a consequence, each input loads its source by a different amount potentially forcing each input to have a different voltage for logic-0 and logic-1. This makes the problem of uneven step sizes worse and may even result in steps getting smaller when they should get bigger as shown in Figure 4c.

As an aside, there's a great word used in DAC theory. Figures 4a and 4b are said to have *monotinicity* because the steps always go up in voltage. Figure 4c does not have full monotinicity.

R/2R ladder network DAC

The R/2R ladder network DAC was designed to overcome the shortcomings of the summing amp DAC. The circuit of an R/2R ladder network DAC is shown in Figure 5.

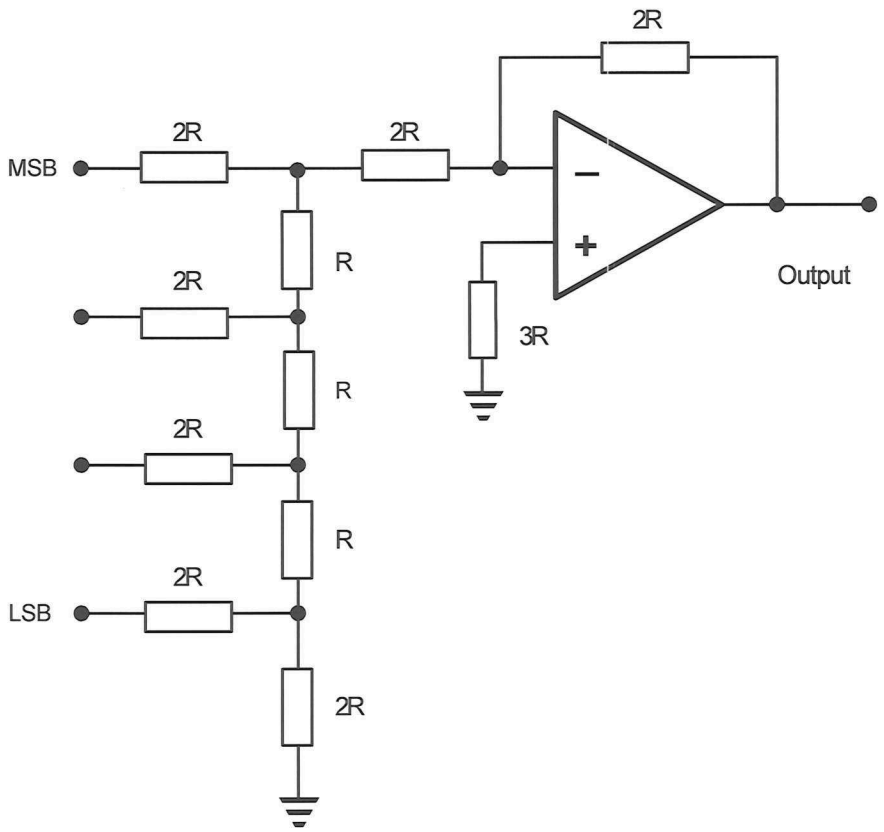


Figure 5 *R/2R ladder network DAC*

This circuit uses only two different values of resistor: *R* and *2R*. In other words, if the resistors labelled *R* were 10kΩ then all the resistors labelled *2R* are 20kΩ. This design makes it a little easier to ensure that all the resistors are exactly the right value. That being the case, the step voltages are more likely to be even.

Helping this issue is the fact that each input's resistance is exactly the same (3×R - but you're not expected to be able to calculate that). With each input having the same input resistance they will load their source by the same amount ensuring that the voltages to each input for a logic-0 and logic-1 will be the same. For these reasons the R/2R DAC is the most common type used inside integrated circuits.

Appendix 6

Digital IC data

The data herein is National Semiconductor's copyrighted material.

Copyright © 2003 National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, CA 95051, U.S.A

All rights reserved.

The data has been reprinted here with permission from National Semiconductor Corporation (http://www.national.com).

National Semiconductor

March 1988

CD4017BM/CD4017BC Decade Counter/Divider with 10 Decoded Outputs
CD4022BM/CD4022BC Divide-by-8 Counter/Divider with 8 Decoded Outputs

General Description

The CD4017BM/CD4017BC is a 5-stage divide-by-10 Johnson counter with 10 decoded outputs and a carry out bit.

The CD4022BM/CD4022BC is a 4-stage divide-by-8 Johnson counter with 8 decoded outputs and a carry-out bit.

These counters are cleared to their zero count by a logical "1" on their reset line. These counters are advanced on the positive edge of the clock signal when the clock enable signal is in the logical "0" state.

The configuration of the CD4017BM/CD4017BC and CD4022BM/CD4022BC permits medium speed operation and assures a hazard free counting sequence. The 10/8 decoded outputs are normally in the logical "0" state and go to the logical "1" state only at their respective time slot. Each decoded output remains high for 1 full clock cycle. The carry-out signal completes a full cycle for every 10/8 clock input cycles and is used as a ripple carry signal to any succeeding stages.

Features

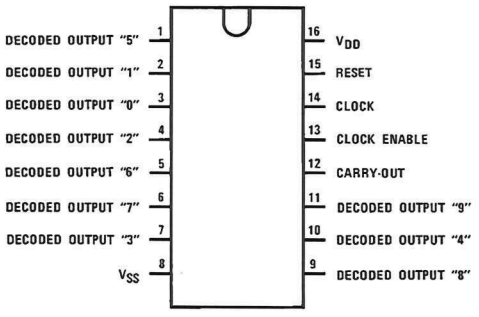
■ Wide supply voltage range	3.0V to 15V
■ High noise immunity	0.45 V_{DD} (typ.)
■ Low power TTL compatibility	Fan out of 2 driving 74L or 1 driving 74LS
■ Medium speed operation	5.0 MHz (typ.) with 10V V_{DD}
■ Low power	10 μW (typ.)
■ Fully static operation	

Applications

- Automotive
- Instrumentation
- Medical electronics
- Alarm systems
- Industrial electronics
- Remote metering

Connection Diagrams

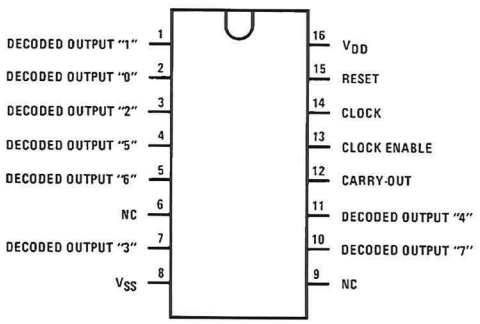
CD4017B
Dual-In-Line Package



TL/F/5950–1

Top View

CD4022B
Dual-In-Line Package



TL/F/5950–2

Top View

Order Number CD4017B or CD4022B

©1995 National Semiconductor Corporation

RRD-B30M105/Printed in U. S. A.

Absolute Maximum Ratings (Notes 1 & 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

DC Supply Voltage (V_{DD})	$-0.5\ V_{DC}$ to $+18\ V_{DC}$
Input Voltage (V_{IN})	$-0.5\ V_{DC}$ to $V_{DD} + 0.5\ V_{DC}$
Storage Temperature (T_S)	$-65°C$ to $+150°C$
Power Dissipation (P_D)	
Dual-In-Line	700 mW
Small Outline	500 mW
Lead Temperature (T_L) (Soldering, 10 seconds)	260°C

Recommended Operating Conditions (Note 2)

DC Supply Voltage (V_{DD})	$+3\ V_{DC}$ to $+15\ V_{DC}$
Input Voltage (V_{IN})	0 to $V_{DD}\ V_{DC}$
Operating Temperature Range (T_A)	
CD4017BM, CD4022BM	$-55°C$ to $+125°C$
CD4017BC, CD4022BC	$-40°C$ to $+85°C$

DC Electrical Characteristics CD4017BM, CD4022BM (Note 2)

Symbol	Parameter	Conditions	−55°C		+25°			+125°C		Units
			Min	Max	Min	Typ	Max	Min	Max	
I_{DD}	Quiescent Device Current	$V_{DD} = 5V$, $V_{IN} = V_{DD}$ or V_{SS}		5		0.3	5		150	μA
		$V_{DD} = 10V$, $V_{IN} = V_{DD}$ or V_{SS}		10		0.5	10		300	μA
		$V_{DD} = 15V$, $V_{IN} = V_{DD}$ or V_{SS}		20		1.0	20		600	μA
V_{OL}	Low Level Output Voltage	$\|I_O\| < 1.0\ \mu A$								
		$V_{DD} = 5V$		0.05		0	0.05		0.05	V
		$V_{DD} = 10V$		0.05		0	0.05		0.05	V
		$V_{DD} = 15V$		0.05		0	0.05		0.05	V
V_{OH}	High Level Output Voltage	$\|I_O\| < 1.0\ \mu A$								
		$V_{DD} = 5V$	4.95		4.95	5		4.95		V
		$V_{DD} = 10V$	9.95		9.95	10		9.95		V
		$V_{DD} = 15V$	14.95		14.95	15		14.95		V
V_{IL}	Low Level Input Voltage	$\|I_O\| < 1.0\ \mu A$								
		$V_{DD} = 5V$, $V_O = 0.5V$ or $4.5V$		1.5			1.5		1.5	V
		$V_{DD} = 10V$, $V_O = 1.0V$ or $9.0V$		3.0			3.0		3.0	V
		$V_{DD} = 15V$, $V_O = 1.5V$ or $13.5V$		4.0			4.0		4.0	V
V_{IH}	High Level Input Voltage	$\|I_O\| < 1.0\ \mu A$								
		$V_{DD} = 5V$, $V_O = 0.5V$ or $4.5V$	3.5		3.5			3.5		V
		$V_{DD} = 10V$, $V_O = 1.0V$ or $9.0V$	7.0		7.0			7.0		V
		$V_{DD} = 15V$, $V_O = 1.5V$ or $13.5V$	11.0		11.0			11.0		V
I_{OL}	Low Level Output Current (Note 3)	$V_{DD} = 5V$, $V_O = 0.4V$	0.64		0.51	0.88		0.36		mA
		$V_{DD} = 10V$, $V_O = 0.5V$	1.6		1.3	2.25		0.9		mA
		$V_{DD} = 15V$, $V_O = 1.5V$	4.2		3.4	8.8		2.4		mA
I_{OH}	High Level Output Current (Note 3)	$V_{DD} = 5V$, $V_O = 4.6V$	−0.25		−0.2	−0.36		−0.14		mA
		$V_{DD} = 10V$, $V_O = 9.5V$	−0.62		−0.5	−0.9		−0.35		mA
		$V_{DD} = 15V$, $V_O = 13.5V$	−1.8		−1.5	−3.5		−1.1		mA
I_{IN}	Input Current	$V_{DD} = 15V$, $V_{IN} = 0V$		−0.1		-10^{-5}	−0.1		−1.0	μA
		$V_{DD} = 15V$, $V_{IN} = 15V$		0.1		10^{-5}	0.1		1.0	μA

DC Electrical Characteristics CD4017BC, CD4022BC (Note 2)

Symbol	Parameter	Conditions	−40°C		+25°			+85°C		Units
			Min	Max	Min	Typ	Max	Min	Max	
I_{DD}	Quiescent Device Current	$V_{DD} = 5V$		20		0.5	20		150	μA
		$V_{DD} = 10V$		40		1.0	40		300	μA
		$V_{DD} = 15V$		80		5.0	80		600	μA
V_{OL}	Low Level Output Voltage	$\|I_O\| < 1.0\ \mu A$								
		$V_{DD} = 5V$		0.05		0	0.05		0.05	V
		$V_{DD} = 10V$		0.05		0	0.05		0.05	V
		$V_{DD} = 15V$		0.05		0	0.05		0.05	V
V_{OH}	High Level Output Voltage	$\|I_O\| < 1.0\ \mu A$								
		$V_{DD} = 5V$	4.95		4.95	5		4.95		V
		$V_{DD} = 10V$	9.95		9.95	10		9.95		V
		$V_{DD} = 15V$	14.95		14.95	15		14.95		V

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed, they are not meant to imply that the devices should be operated at these limits. The table of "Recommended Operating Conditions" and "Electrical Characteristics" provides conditions for actual device operation.

Note 2: $V_{SS} = 0V$ unless otherwise specified.

Note 3: I_{OL} and I_{OH} are tested one output at a time.

National Semiconductor

February 1988

CD4029BM/CD4029BC Presettable Binary/Decade Up/Down Counter

General Description

The CD4029BM/CD4029BC is a presettable up/down counter which counts in either binary or decade mode depending on the voltage level applied at binary/decade input. When binary/decade is at logical "1", the counter counts in binary, otherwise it counts in decade. Similarly, the counter counts up when the up/down input is at logical "1" and vice versa.

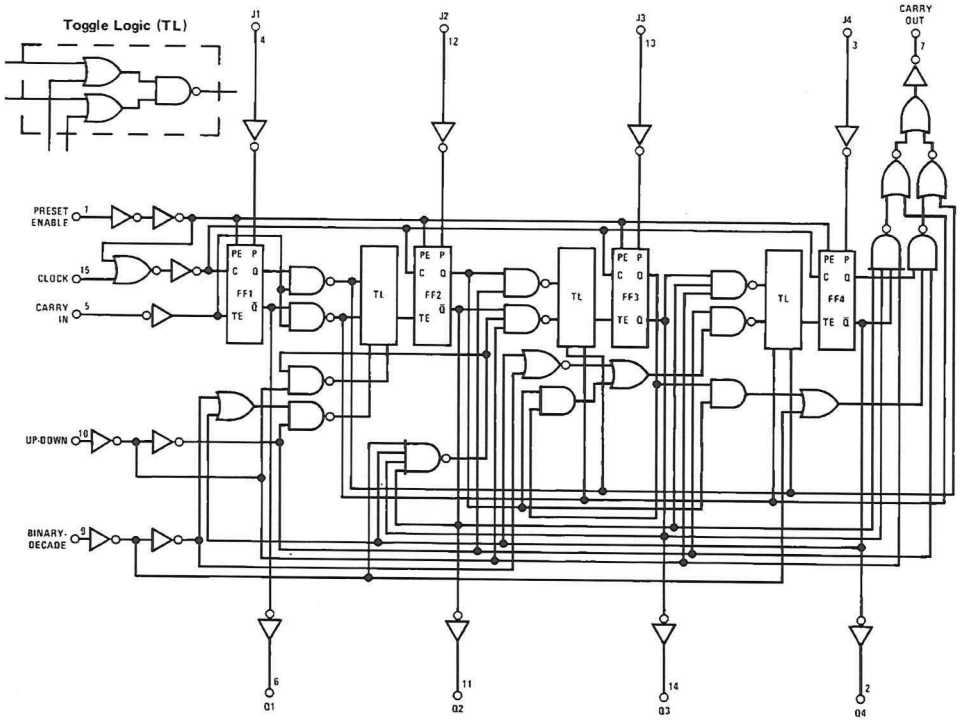
A logical "1" preset enable signal allows information at the "jam" inputs to preset the counter to any state asynchronously with the clock. The counter is advanced one count at the positive-going edge of the clock if the carry in and preset enable inputs are at logical "0". Advancement is inhibited when either or both of these two inputs is at logical "1". The carry out signal is normally at logical "1" state and goes to logical "0" state when the counter reaches its maximum count in the "up" mode or the minimum count in the "down" mode provided the carry input is at logical "0" state.

All inputs are protected against static discharge by diode clamps to both V_{DD} and V_{SS}.

Features

- Wide supply voltage range — 3V to 15V
- High noise immunity — 0.45 V_{DD} (typ.)
- Low power TTL compatibility — fan out of 2 driving 74L or 1 driving 74LS
- Parallel jam inputs
- Binary or BCD decade up/down counting

Logic Diagram



TL/F/5960–1

©1995 National Semiconductor Corporation

RRD-B30M105/Printed in U. S. A.

Absolute Maximum Ratings

(Notes 1 and 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

DC Supply Voltage (V_{DD})	$-0.5V$ to $+18$ V_{DC}
Input Voltage (V_{IN})	$-0.5V$ to $V_{DD} + 0.5$ V_{DC}
Storage Temperature Range (T_S)	$-65°C$ to $+150°C$
Power Dissipation (P_D)	
Dual-In-Line	700 mW
Small Outline	500 mW
Lead Temperature (T_L)	
(Soldering, 10 seconds)	260°C

Recommended Operating Conditions (Note 2)

DC Supply Voltage (V_{DD})	3V to 15 V_{DC}
Input Voltage (V_{IN})	0V to V_{DD} V_{DC}
Operating Temperature Range (T_A)	
CD4029BM	$-55°C$ to $+125°C$
CD4029BC	$-40°C$ to $+85°C$

DC Electrical Characteristics CD4029BM (Note 2)

Symbol	Parameter	Conditions	−55°C		+25°C			+125°C		Units
			Min	Max	Min	Typ	Max	Min	Max	
I_{DD}	Quiescent Device Current	$V_{DD} = 5V$		5			5	5	150	μA
		$V_{DD} = 10V$		10			10		300	μA
		$V_{DD} = 15V$		20			20		600	μA
V_{OL}	Low Level Output Voltage	$\|I_O\| < 1$ μA								
		$V_{DD} = 5V$		0.05		0	0.05		0.05	V
		$V_{DD} = 10V$		0.05		0	0.05		0.05	V
		$V_{DD} = 15V$		0.05		0	0.05		0.05	V
V_{OH}	High Level Output Voltage	$\|I_O\| < 1$ μA								
		$V_{DD} = 5V$	4.95		4.95	5		4.95		V
		$V_{DD} = 10V$	9.95		9.95	10		9.95		V
		$V_{DD} = 15V$	14.95		14.95	15		14.95		V
V_{IL}	Low Level Input Voltage	$V_{DD} = 5V$, $V_O = 0.5V$ or $4.5V$		1.5			1.5		1.5	V
		$V_{DD} = 10V$, $V_O = 1V$ or $9V$		3.0			3.0		3.0	V
		$V_{DD} = 15V$, $V_O = 1.5V$ or $13.5V$		4.0			4.0		4.0	V
V_{IH}	High Level Input Voltage	$V_{DD} = 5V$, $V_O = 0.5V$ or $4.5V$	3.5		3.5			3.5		V
		$V_{DD} = 10V$, $V_O = 1V$ or $9V$	7.0		7.0			7.0		V
		$V_{DD} = 15V$, $V_O = 1.5V$ or $13.5V$	11.0		11.0			11.0		V
I_{OL}	Low Level Output Current (Note 3)	$V_{DD} = 5V$, $V_O = 0.4V$	0.64		0.51	0.88		0.36		mA
		$V_{DD} = 10V$, $V_O = 0.5V$	1.6		1.3	2.25		0.9		mA
		$V_{DD} = 15V$, $V_O = 1.5V$	4.2		3.4	8.8		2.4		mA
I_{OH}	High Level Output Current (Note 3)	$V_{DD} = 5V$, $V_O = 4.6V$	−0.64		−0.51	−0.88		−0.36		mA
		$V_{DD} = 10V$, $V_O = 9.5V$	−1.6		−1.3	−2.25		−0.9		mA
		$V_{DD} = 15V$, $V_O = 13.5V$	−4.2		−3.4	−8.8		−2.4		mA
I_{IN}	Input Current	$V_{DD} = 15V$, $V_{IN} = 0V$		−0.1		-10^{-5}	−0.1		−1.0	μA
		$V_{DD} = 15V$, $V_{IN} = 15V$		0.1		10^{-5}	0.1		1.0	μA

DC Electrical Characteristics CD4029BC (Note 2)

Symbol	Parameter	Conditions	−40°C		+25°C			+85°C		Units
			Min	Max	Min	Typ	Max	Min	Max	
I_{DD}	Quiescent Device Current	$V_{DD} = 5V$		20			20		150	μA
		$V_{DD} = 10V$		40			40		300	μA
		$V_{DD} = 15V$		80			80		600	μA
V_{OL}	Low Level Output Voltage	$\|I_O\| < 1$ μA								
		$V_{DD} = 5V$		0.05		0	0.05		0.05	V
		$V_{DD} = 10V$		0.05		0	0.05		0.05	V
		$V_{DD} = 15V$		0.05		0	0.05		0.05	V
V_{OH}	High Level Output Voltage	$\|I_O\| < 1$ μA								
		$V_{DD} = 5V$	4.95		4.95	5		4.95		V
		$V_{DD} = 10V$	9.95		9.95	10		9.95		V
		$V_{DD} = 15V$	14.95		14.95	15		14.95		V

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. Except for "Operating Temperature Range" they are not meant to imply that the devices should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: $V_{SS} = 0V$ unless otherwise specified.

Note 3: I_{OH} and I_{OL} are tested one output at a time.

National Semiconductor

August 1995

DM74ALS00A
Quad 2-Input NAND Gate

General Description

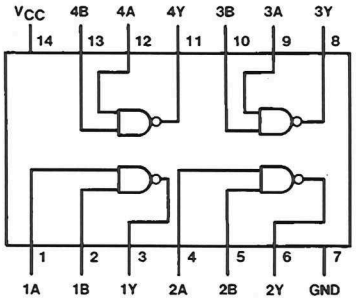
This device contains four independent gates, each of which performs the logic NAND function.

Features

- Switching specifications at 50 pF
- Switching specifications guaranteed over full temperature and V_{CC} range
- Advanced oxide-isolated, ion-implanted Schottky TTL process
- Functionally and pin for pin compatible with Schottky and low power Schottky TTL counterpart
- Improved AC performance over Schottky and low power Schottky counterparts

Connection Diagram

Dual-In-Line Package



TL/F/6270–1

Order Number DM74ALS00M, DM74ALS00SJ or DM74ALS00N
See NS Package Number M14A, M14D or N14A

Function Table

$Y = \overline{AB}$

Inputs		Output
A	B	Y
L	L	H
L	H	H
H	L	H
H	H	L

H = High Logic Level
L = Low Logic Level

©1995 National Semiconductor Corporation TL/F/6270

RRD-B30M115/Printed in U. S. A.

Absolute Maximum Ratings

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
DM74ALS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C
Typical θ_{JA}	
N Package	86.5°C/W
M Package	116.0°C/W

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter	DM74ALS00A			Units
		Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	V
V_{IH}	High Level Input Voltage	2			V
V_{IL}	Low Level Input Voltage			0.8	V
I_{OH}	High Level Output Current			−0.4	mA
I_{OL}	Low Level Output Current			8	mA
T_A	Free Air Operating Temperature	0		70	°C

Electrical Characteristics

over recommended operating free air temperature range. All typical values are measured at V_{CC} = 5V, T_A = 25°C.

Symbol	Parameter	Conditions		Min	Typ	Max	Units
V_{IK}	Input Clamp Voltage	V_{CC} = 4.5V, I_I = −18 mA				−1.5	V
V_{OH}	High Level Output Voltage	I_{OH} = −0.4 mA V_{CC} = 4.5V to 5.5V		V_{CC} − 2			V
V_{OL}	Low Level Output Voltage	V_{CC} = 4.5V	74ALS I_{OL} = 8 mA		0.35	0.5	V
I_I	Input Current at Max Input Voltage	V_{CC} = 5.5V, V_{IH} = 7V				0.1	mA
I_{IH}	High Level Input Current	V_{CC} = 5.5V, V_{IH} = 2.7V				20	μA
I_{IL}	Low Level Input Current	V_{CC} = 5.5V, V_{IL} = 0.4V				−0.1	mA
I_O	Output Drive Current	V_{CC} = 5.5V	V_O = 2.25V	−30		−112	mA
I_{CC}	Supply Current	V_{CC} = 5.5V	Outputs High		0.43	0.85	mA
			Outputs Low		1.62	3	mA

Switching Characteristics over recommended operating free air temperature range (Note 1)

Symbol	Parameter	Conditions	DM74ALS00A		Units
			Min	Max	
t_{PLH}	Propagation Delay Time Low to High Level Output	V_{CC} = 4.5V to 5.5V R_L = 500Ω C_L = 50 pF	3	11	ns
t_{PHL}	Propagation Delay Time High to Low Level Output		2	8	ns

Note 1: See Section 5 for test waveforms and output load.

National Semiconductor

August 1995

DM64ALS08/DM74ALS08 Quad 2-Input AND Gate

General Description

This device contains four independent gates, each of which performs the logic AND function.

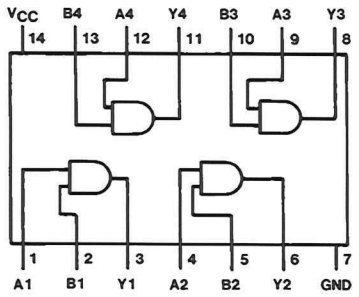
The DM64ALS08 version features the same performance as the standard version DM74ALS08 with a guarantee over an extended temperature range (−40°C to +85°C).

Features

- Switching specifications at 50 pF
- Switching specifications guaranteed over full temperature and V_{CC} range
- Advanced oxide-isolated, ion-implanted Schottky TTL process
- Functionally and pin for pin compatible with Schottky and low power Schottky TTL counterpart
- Improved AC performance over Schottky and low power Schottky counterparts
- DM64ALS08 guaranteed over extended temperature −40°C to +85°C

Connection Diagram

Dual-In-Line Package



TL/F/6271–1

Order Number DM64ALS08M, DM64ALS08N, DM74ALS08M, DM74ALS08N or DM74ALS08SJ
See NS Package Number M14A, M14D or N14A

Function Table

$Y = AB$

Inputs		Output
A	B	Y
L	L	L
L	H	L
H	L	L
H	H	H

H = High Logic Level
L = Low Logic Level

©1995 National Semiconductor Corporation

RRD-B30M115/Printed in U. S. A.

Absolute Maximum Ratings

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
DM74ALS	0°C to +70°C
DM64ALS	−40°C to +85°C
Storage Temperature Range	−65°C to +150°C
Typical θ_{JA}	
N Package	89°C/W
M Package	120°C/W

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter	DM64ALS08			DM74ALS08			Units
		Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	4.5	5	5.5	V
V_{IH}	High Level Input Voltage	2			2			V
V_{IL}	Low Level Input Voltage			0.8			0.8	V
I_{OH}	High Level Output Current			−0.4			−0.4	mA
I_{OL}	Low Level Output Current			8			8	mA
T_A	Free Air Operating Temperature	−40		85	0		70	°C

Electrical Characteristics

over recommended operating free air temperature range. All typical values are measured at V_{CC} = 5V, T_A = 25°C.

Symbol	Parameter	Conditions		Min	Typ	Max	Units
V_{IK}	Input Clamp Voltage	V_{CC} = 4.5V, I_I = −18 mA				−1.5	V
V_{OH}	High Level Output Voltage	I_{OH} = −0.4 mA V_{CC} = 4.5V to 5.5V		V_{CC} − 2			V
V_{OL}	Low Level Output Voltage	V_{CC} = 4.5V	I_{OL} = 4 mA		0.25	0.4	V
			I_{OL} = 8 mA		0.35	0.5	V
I_I	Input Current @ Max. Input Voltage	V_{CC} = 5.5V, V_{IH} = 7V				0.1	mA
I_{IH}	High Level Input Current	V_{CC} = 5.5V, V_{IH} = 2.7V				20	μA
I_{IL}	Low Level Input Current	V_{CC} = 5.5V, V_{IL} = 0.4V				−0.1	mA
I_O	Output Drive Current	V_{CC} = 5.5V	V_O = 2.25V	−30		−112	mA
I_{CC}	Supply Current	V_{CC} = 5.5V	Outputs High		1.3	2.4	mA
			Outputs Low		2.2	4	mA

Switching Characteristics

over recommended operating free air temperature range (Note 1).

Symbol	Parameter	Conditions	DM64ALS08		DM74ALS08		Units
			Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time Low to High Level Output	V_{CC} = 4.5V to 5.5V R_L = 500Ω C_L = 50 pF	4	14	4	14	ns
t_{PHL}	Propagation Delay Time High to Low Level Output		3	10	3	10	ns

Note 1: See Section 5 for test waveforms and output load.

National Semiconductor

July 1992

DM54LS47/DM74LS47 BCD to 7-Segment Decoder/Driver with Open-Collector Outputs

General Description

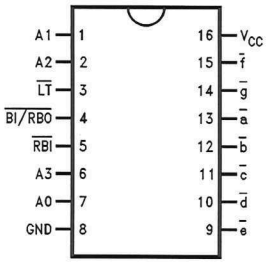
The 'LS47 accepts four lines of BCD (8421) input data, generates their complements internally and decodes the data with seven AND/OR gates having open-collector outputs to drive indicator segments directly. Each segment output is guaranteed to sink 24 mA in the ON (LOW) state and withstand 15V in the OFF (HIGH) state with a maximum leakage current of 250 μA. Auxiliary inputs provided blanking, lamp test and cascadable zero-suppression functions.

Features

- Open-collector outputs
- Drive indicator segments directly
- Cascadable zero-suppression capability
- Lamp test input

Connection Diagram

Dual-In-Line Package



TL/F/9817–1

Order Number DM54LS47J, DM54LS47W, DM74LS47M or DM74LS47N
See NS Package Number J16A, M16A, N16E or W16A

Pin Names	Description
A0–A3	BCD Inputs
$\overline{RBI}$	Ripple Blanking Input (Active LOW)
$\overline{LT}$	Lamp Test Input (Active LOW)
$\overline{BI/RBO}$	Blanking Input (Active LOW) or Ripple Blanking Output (Active LOW)
$\overline{a}$–$\overline{g}$	*Segment Outputs (Active LOW)

*OC—Open Collector

©1995 National Semiconductor Corporation

RRD-B30M115/Printed in U. S. A.

Switching Characteristics at V_{CC} = +5.0V, T_A = +25°C

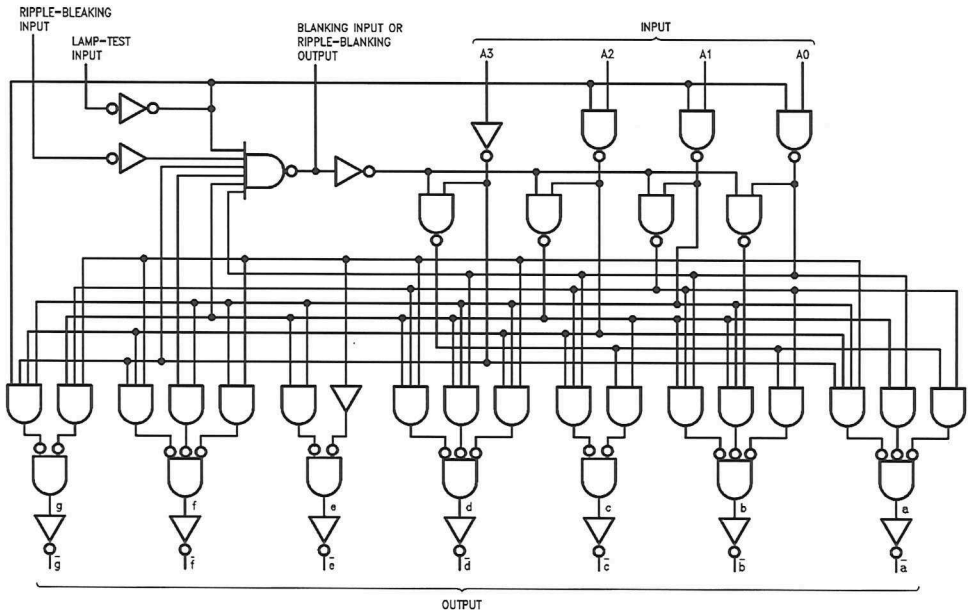
Symbol	Parameter	Conditions	R_L = 665Ω, C_L = 15 pF		Units
			Min	Max	
t_{PLH} t_{PHL}	Propagation Delay An to $\overline{a}$–$\overline{g}$			100 100	ns
t_{PLH} t_{PHL}	Propagation Delay $\overline{RBI}$ to $\overline{a}$–$\overline{g}$*			100 100	ns

*$\overline{LT}$ = HIGH, A0–A3 = LOW

Functional Description

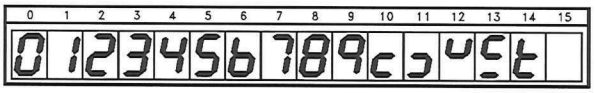
The 'LS47 decodes the input data in the pattern indicated in the Truth Table and the segment identification illustration. If the input data is decimal zero, a LOW signal applied to the $\overline{RBI}$ blanks the display and causes a multidigit display. For example, by grounding the $\overline{RBI}$ of the highest order decoder and connecting its $\overline{BI/RBO}$ to $\overline{RBI}$ of the next lowest order decoder, etc., leading zeros will be suppressed. Similarly, by grounding $\overline{RBI}$ of the lowest order decoder and connecting its $\overline{BI/RBO}$ to $\overline{RBI}$ of the next highest order decoder, etc., trailing zeros will be suppressed. Leading and trailing zeros can be suppressed simultaneously by using external gates, i.e.: by driving $\overline{RBI}$ of a intermediate decoder from an OR gate whose inputs are $\overline{BI/RBO}$ of the next highest and lowest order decoders. $\overline{BI/RBO}$ also serves as an unconditional blanking input. The internal NAND gate that generates the $\overline{RBO}$ signal has a resistive pull-up, as opposed to a totem pole, and thus $\overline{BI/RBO}$ can be forced LOW by external means, using wired-collector logic. A LOW signal thus applied to $\overline{BI/RBO}$ turns off all segment outputs. This blanking feature can be used to control display intensity by varying the duty cycle of the blanking signal. A LOW signal applied to $\overline{LT}$ turns on all segment outputs, provided that $\overline{BI/RBO}$ is not forced LOW.

Logic Diagram



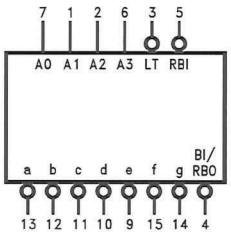
TL/F/9817–3

Numerical Designations—Resultant Displays



TL/F/9817–4

Logic Symbol



TL/F/9817–2

V_{CC} = Pin 16
GND = Pin 8

Truth Table

Decimal or Function	Inputs							Outputs							Note
	$\overline{LT}$	$\overline{RBI}$	A3	A2	A1	A0	$\overline{BI/RBO}$	$\overline{a}$	$\overline{b}$	$\overline{c}$	$\overline{d}$	$\overline{e}$	$\overline{f}$	$\overline{g}$	
0	H	H	L	L	L	L	H	L	L	L	L	L	L	H	1
1	H	X	L	L	L	H	H	H	L	L	H	H	H	H	1
2	H	X	L	L	H	L	H	L	L	H	L	L	H	L	
3	H	X	L	L	H	H	H	L	L	L	L	H	H	L	
4	H	X	L	H	L	L	H	H	L	L	H	H	L	L	
5	H	X	L	H	L	H	H	L	H	L	L	H	L	L	
6	H	X	L	H	H	L	H	H	H	L	L	L	L	L	
7	H	X	L	H	H	H	H	L	L	L	H	H	H	H	
8	H	X	H	L	L	L	H	L	L	L	L	L	L	L	
9	H	X	H	L	L	H	H	L	L	L	H	H	L	L	
10	H	X	H	L	H	L	H	H	H	H	L	L	H	L	
11	H	X	H	L	H	H	H	H	H	L	L	H	H	L	
12	H	X	H	H	L	L	H	H	L	H	H	H	L	L	
13	H	X	H	H	L	H	H	L	H	H	L	H	L	L	
14	H	X	H	H	H	L	H	H	H	H	L	L	L	L	
15	H	X	H	H	H	H	H	H	H	H	H	H	H	H	
$\overline{BI}$	X	X	X	X	X	X	L	H	H	H	H	H	H	H	2
$\overline{RBI}$	H	L	L	L	L	L	L	H	H	H	H	H	H	H	3
$\overline{LT}$	L	X	X	X	X	X	H	L	L	L	L	L	L	L	4

Note 1: $\overline{BI/RBO}$ is wire-AND logic serving as blanking input ($\overline{BI}$) and/or ripple-blanking output ($\overline{RBO}$). The blanking out ($\overline{BI}$) must be open or held at a HIGH level when output functions 0 through 15 are desired, and ripple-blanking input ($\overline{RBI}$) must be open or at a HIGH level if blanking or a decimal 0 is not desired. X = input may be HIGH or LOW.

Note 2: When a LOW level is applied to the blanking input (forced condition) all segment outputs go to a HIGH level regardless of the state of any other input condition.

Note 3: When ripple-blanking input ($\overline{RBI}$) and inputs A0, A1, A2 and A3 are LOW level, with the lamp test input at HIGH level, all segment outputs go to a HIGH level and the ripple-blanking output ($\overline{RBO}$) goes to a LOW level (response condition).

Note 4: When the blanking input/ripple-blanking output ($\overline{BI/RBO}$) is open or held at a HIGH level, and a LOW level is applied to lamp test input, all segment outputs go to a LOW level.

National Semiconductor

May 1989

DM74LS122 Retriggerable One-Shot with Clear and Complementary Outputs

General Description

The DM74LS122 is a retriggerable monostable multivibrator featuring both positive and negative edge triggering with complementary outputs. An internal 10 kΩ timing resistor is provided for design convenience minimizing component count and layout problems. This device can be used with a single external capacitor. The 'LS122 has two active-low transition triggering inputs (A), two active-high transition triggering inputs (B), and a CLEAR input that terminates the output pulse width at a predetermined time independent of the timing components. The clear (CLR) input also serves as a trigger input when it is pulsed with a low level pulse transition (⌝⌞). To obtain optimum and trouble free operation please read operating rules and NSC one-shot application notes carefully and observe recommendations.

Features

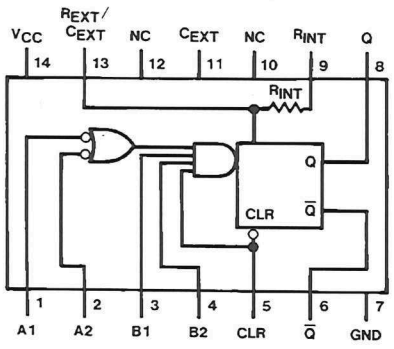
- DC triggered from active-high transition or active-low transition inputs
- Retriggerable to 100% duty cycle
- Over-riding clear terminates output pulse
- Internal 10 kΩ timing resistor
- TTL, DTL compatible
- Compensated for V_{CC} and temperature variations
- Input clamp diodes

Functional Description

The basic output pulse width is determined by selection of the internal resistor R_{INT} or an external resistor (R_X) and capacitor (C_X). Once triggered, the output pulse width may be extended by retriggering the gated active-low (A) transition inputs or the active-high transition (B) inputs or the CLEAR input. The output pulse width can be reduced or terminated by overriding it with the active-low CLEAR input.

Connection Diagram

Dual-In-Line Package



TL/F/6385–1

Order Number DM74LS122M or DM74LS122N
See NS Package Number M14A or N14A

Function Table

Inputs					Outputs	
CLEAR	A1	A2	B1	B2	Q	$\overline{Q}$
L	X	X	X	X	L	H
X	H	H	X	X	L	H
X	X	X	L	X	L	H
X	X	X	X	L	L	H
H	L	X	↑	H	⌞⌜	⌝⌞
H	L	X	H	↑	⌞⌜	⌝⌞
H	X	L	↑	H	⌞⌜	⌝⌞
H	X	L	H	↑	⌞⌜	⌝⌞
H	H	↓	H	H	⌞⌜	⌝⌞
H	↓	↓	H	H	⌞⌜	⌝⌞
H	↓	H	H	H	⌞⌜	⌝⌞
↑	L	X	H	H	⌞⌜	⌝⌞
↑	X	L	H	H	⌞⌜	⌝⌞

H = High Logic Level
L = Low Logic Level
X = Can Be Either Low or High
↑ = Positive Going Transition
↓ = Negative Going Transition
⌞⌜ = A Positive Pulse
⌝⌞ = A Negative Pulse

DM74LS122 Retriggerable One-Shot with Clear and Complementary Outputs

©1995 National Semiconductor Corporation

RRD-B30M105/Printed in U. S. A.

Absolute Maximum Ratings (Note)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range DM74LS	0°C to +70°C
Storage Temperature	−65°C to +150°C

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameters		Min	Nom	Max	Units
V_{CC}	Supply Voltage		4.75	5	5.25	V
V_{IH}	High Level Input Voltage		2			V
V_{IL}	Low Level Input Voltage				0.8	V
I_{OH}	High Level Output Current				−0.4	mA
I_{OL}	Low Level Output Current				8	mA
t_W	Pulse Width (Note 6)	A or B High	40			ns
		A or B Low	40			
		Clear Low	40			
R_{EXT}	External Timing Resistor		5		260	kΩ
C_{EXT}	External Timing Capacitance			No Restriction		μF
C_{WIRE}	Wiring Capacitance at R_{EXT}/C_{EXT} Terminal				50	pF
T_A	Free Air Operating Temperature		0		70	°C

Electrical Characteristics over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I = −18 mA			−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max V_{IL} = Max, V_{IH} = Min	2.7	3.4		V
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max V_{IL} = Max, V_{IH} = Min		0.35	0.5	V
		I_{OL} = 4 mA, V_{CC} = Min		0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V			0.1	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V			20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V			−0.4	mA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 2)	−20		−100	mA
I_{CC}	Supply Current	V_{CC} = Max (Notes 3, 4 and 5)		6	11	mA

National Semiconductor

June 1989

54LS138/DM54LS138/DM74LS138, 54LS139/DM54LS139/DM74LS139 Decoders/Demultiplexers

General Description

These Schottky-clamped circuits are designed to be used in high-performance memory-decoding or data-routing applications, requiring very short propagation delay times. In high-performance memory systems these decoders can be used to minimize the effects of system decoding. When used with high-speed memories, the delay times of these decoders are usually less than the typical access time of the memory. This means that the effective system delay introduced by the decoder is negligible.

The LS138 decodes one-of-eight lines, based upon the conditions at the three binary select inputs and the three enable inputs. Two active-low and one active-high enable inputs reduce the need for external gates or inverters when expanding. A 24-line decoder can be implemented with no external inverters, and a 32-line decoder requires only one inverter. An enable input can be used as a data input for demultiplexing applications.

The LS139 comprises two separate two-line-to-four-line decoders in a single package. The active-low enable input can be used as a data line in demultiplexing applications.

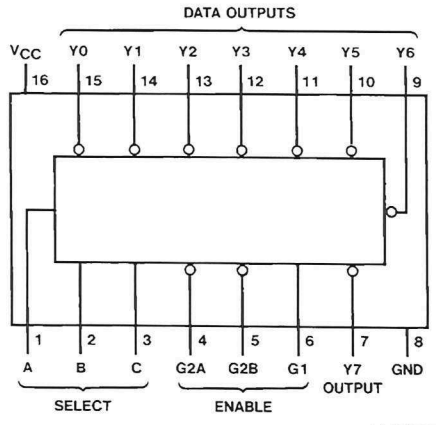
All of these decoders/demultiplexers feature fully buffered inputs, presenting only one normalized load to its driving circuit. All inputs are clamped with high-performance Schottky diodes to suppress line-ringing and simplify system design.

Features

- Designed specifically for high speed:
  - Memory decoders
  - Data transmission systems
- LS138 3-to-8-line decoders incorporates 3 enable inputs to simplify cascading and/or data reception
- LS139 contains two fully independent 2-to-4-line decoders/demultiplexers
- Schottky clamped for high performance
- Typical propagation delay (3 levels of logic)
  - LS138 21 ns
  - LS139 21 ns
- Typical power dissipation
  - LS138 32 mW
  - LS139 34 mW
- Alternate Military/Aerospace devices (54LS138, 54LS139) are available. Contact a National Semiconductor Sales Office/Distributor for specifications.

Connection Diagrams

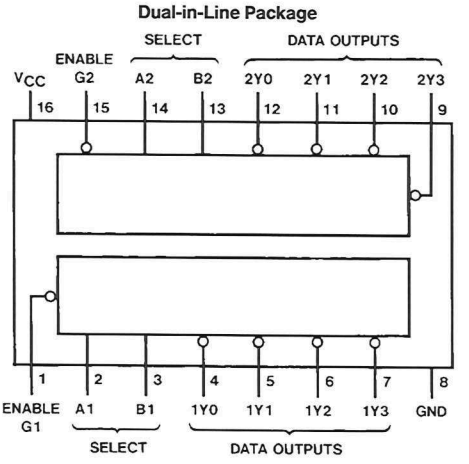
Dual-in-Line Package



TL/F/6391–1

Order Number 54LS138DMQB, 54LS138FMQB, 54LS138LMQB, DM54LS138J, DM54LS138W, DM74LS138M or DM74LS138N
See NS Package Number E20A, J16A, M16A, N16E or W16A

Dual-in-Line Package


SELECT
DATA OUTPUTS
ENABLE
VCC G2 A2 B2 2Y0 2Y1 2Y2 2Y3
16 15 14 13 12 11 10 9
1 2 3 4 5 6 7 8
ENABLE G1 A1 B1 1Y0 1Y1 1Y2 1Y3 GND
SELECT
DATA OUTPUTS

TL/F/6391–2

Order Number 54LS139DMQB, 54LS139FMQB, 54LS139LMQB, DM54LS139J, DM54LS139W, DM74LS139M or DM74LS139N
See NS Package Number E20A, J16A, M16A, N16E or W16A

©1995 National Semiconductor Corporation

RRD-B30M105/Printed in U. S. A.

Absolute Maximum Ratings (Note)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
DM54LS and 54LS	−55°C to +125°C
DM74LS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter	DM54LS138			DM74LS138			Units
		Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High Level Input Voltage	2			2			V
V_{IL}	Low Level Input Voltage			0.7			0.8	V
I_{OH}	High Level Output Current			−0.4			−0.4	mA
I_{OL}	Low Level Output Current			4			8	mA
T_A	Free Air Operating Temperature	−55		125	0		70	°C

'LS138 Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions		Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA				−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max, V_{IH} = Min	DM54	2.5	3.4		V
			DM74	2.7	3.4		
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IL} = Max, V_{IH} = Min	DM54		0.25	0.4	V
			DM74		0.35	0.5	
		I_{OL} = 4 mA, V_{CC} = Min	DM74		0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V				0.1	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V				20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V				−0.36	mA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 2)	DM54	−20		−100	mA
			DM74	−20		−100	
I_{CC}	Supply Current	V_{CC} = Max (Note 3)			6.3	10	mA

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Note 3: I_{CC} is measured with all outputs enabled and open.

'LS138 Switching Characteristics

at V_{CC} = 5V and T_A = 25°C (See Section 1 for Test Waveforms and Output Load)

Symbol	Parameter	From (Input) To (Output)	Levels of Delay	R_L = 2 kΩ				Units
				C_L = 15 pF		C_L = 50 pF		
				Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time Low to High Level Output	Select to Output	2		18		27	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Select to Output	2		27		40	ns
t_{PLH}	Propagation Delay Time Low to High Level Output	Select to Output	3		18		27	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Select to Output	3		27		40	ns
t_{PLH}	Propagation Delay Time Low to High Level Output	Enable to Output	2		18		27	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Enable to Output	2		24		40	ns
t_{PLH}	Propagation Delay Time Low to High Level Output	Enable to Output	3		18		27	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Enable to Output	3		28		40	ns

Recommended Operating Conditions

Symbol	Parameter	DM54LS139			DM74LS139			Units
		Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High Level Input Voltage	2			2			V
V_{IL}	Low Level Input Voltage			0.7			0.8	V
I_{OH}	High Level Output Current			−0.4			−0.4	mA
I_{OL}	Low Level Output Current			4			8	mA
T_A	Free Air Operating Temperature	−55		125	0		70	°C

'LS139 Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions		Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA				−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max, V_{IH} = Min	DM54	2.5	3.4		V
			DM74	2.7	3.4		
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max V_{IL} = Max, V_{IH} = Min	DM54		0.25	0.4	V
			DM74		0.35	0.5	
		I_{OL} = 4 mA, V_{CC} = Min	DM74		0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V				0.1	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V				20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V				−0.36	mA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 2)	DM54	−20		−100	mA
			DM74	−20		−100	
I_{CC}	Supply Current	V_{CC} = Max (Note 3)			6.8	11	mA

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Note 3: I_{CC} is measured with all outputs enabled and open.

'LS139 Switching Characteristics

at V_{CC} = 5V and T_A = 25°C (See Section 1 for Test Waveforms and Output Load)

Symbol	Parameter	From (Input) To (Output)	R_L = 2 kΩ				Units
			C_L = 15 pF		C_L = 50 pF		
			Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time Low to High Level Output	Select to Output		18		27	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Select to Output		27		40	ns
t_{PLH}	Propagation Delay Time Low to High Level Output	Enable to Output		18		27	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Enable to Output		24		40	ns

Function Tables

LS138

Inputs					Outputs							
Enable		Select										
G1	G2*	C	B	A	YO	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

* G2 = G2A + G2B

H = High Level, L = Low Level, X = Don't Care

LS139

Inputs			Outputs			
Enable	Select					
G	B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

H = High Level, L = Low Level, X = Don't Care

National Semiconductor

August 1995

DM74ALS151
1 of 8 Line Data Selector/Multiplexer

General Description

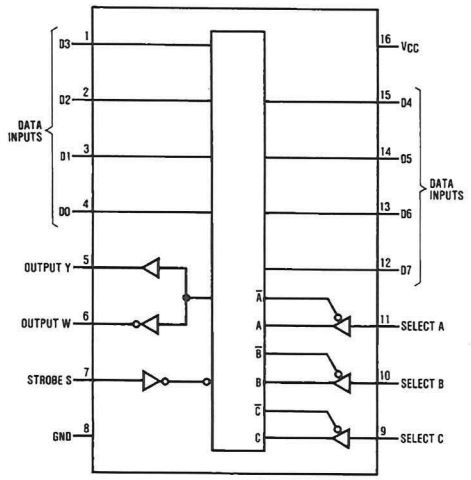
This Data Selector/Multiplexer contains full on-chip decoding to select one-of-eight data sources as a result of a unique three-bit binary code at the Select inputs. Two complementary outputs provide both inverting and non-inverting buffer operation. A Strobe input is provided which, when at the high level, disables all data inputs and forces the Y output to the low state and the W output to the high state. The Select input buffers incorporate internal overlap features to ensure that select input changes do not cause invalid output transients.

Features

- Advanced oxide-isolated, ion-implanted Schottky TTL process
- Switching performance is guaranteed over full temperature and V_{CC} supply range
- Pin and functional compatible with LS family counterpart
- Improved output transient handling capability

Connection Diagram

Dual-In-Line Package



TL/F/6203–1

Order Number DM74ALS151M or DM74ALS151N
See NS Package Number M16A or N16A

Function Table

Inputs				Outputs	
Select			Strobe	Y	W
C	B	A	S		
X	X	X	H	L	H
L	L	L	L	D0	$\overline{D0}$
L	L	H	L	D1	$\overline{D1}$
L	H	L	L	D2	$\overline{D2}$
L	H	H	L	D3	$\overline{D3}$
H	L	L	L	D4	$\overline{D4}$
H	L	H	L	D5	$\overline{D5}$
H	H	L	L	D6	$\overline{D6}$
H	H	H	L	D7	$\overline{D7}$

H = High Level, L = Low Level, X = Don't Care

D0 thru D7 = the level of the respective D input

©1995 National Semiconductor Corporation

RRD-B30M105/Printed in U. S. A.

Absolute Maximum Ratings

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
DM74ALS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C
Typical θ_{JA}	
N Package	78.0°C/W
M Package	107.0°C/W

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter	DM74ALS151			Units
		Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	V
V_{IH}	High Level Input Voltage	2			V
V_{IL}	Low Level Input Voltage			0.8	V
I_{OH}	High Level Output Current			−2.6	mA
I_{OL}	Low Level Output Current			24	mA
T_A	Free Air Operating Temperature	0		70	°C

Electrical Characteristics

over recommended operating free-air temperature range. All typical values are measured at V_{CC} = 5V, T_A = 25°C.

Symbol	Parameter	Conditions		Min	Typ	Max	Units
V_{IK}	Input Clamp Voltage	V_{CC} = 4.5V, I_{IN} = −18 mA				−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = 4.5V, I_{OH} = Max		2.4	3.2		V
		I_{OH} = −400 μA, V_{CC} = 4.5V to 5.5V		V_{CC} − 2			V
V_{OL}	Low Level Output Voltage	V_{CC} = 4.5V	74ALS I_{OL} = 24 mA		0.35	0.5	V
I_I	Input Current at Max Input Voltage	V_{CC} = 5.5V, V_{IN} = 7V				0.1	mA
I_{IH}	High Level Input Current	V_{CC} = 5.5V, V_{IN} = 2.7V				20	μA
I_{IL}	Low Level Input Current	V_{CC} = 5.5V, V_{IN} = 0.4V				−0.1	mA
I_O	Output Drive Current	V_{CC} = 5.5V, V_{OUT} = 2.25V		−30		−112	mA
I_{CC}	Supply Current	V_{CC} = 5.5V All Inputs = 4.5V			7.5	12	mA

National Semiconductor

August 1995

DM74ALS153 Dual 1 of 4 Line Data Selector/Multiplexer

General Description

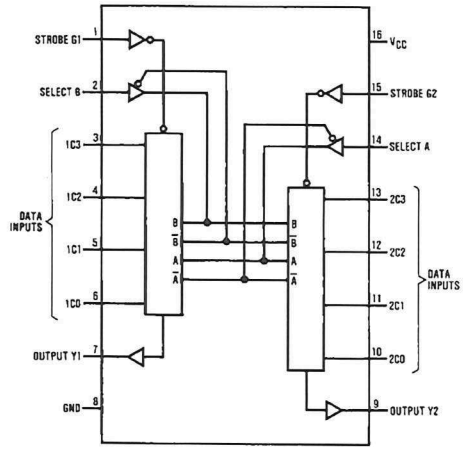
This Data Selector/Multiplexer contains full on-chip decoding to select one-of-four data sources as a result of a unique two-bit binary code at the Select inputs. Each of the two Data Selector/Multiplexer circuits have their own separate Data and Strobe inputs and a non-inverting output buffer. The Select inputs A and B are common to both sections. The Strobe inputs, when at the high level, disable their associated data inputs and force the corresponding output to the low state. The Select input buffers incorporate internal overlap features to ensure that select input changes do not cause invalid output transients.

Features

- Advanced oxide-isolated, ion-implanted Schottky TTL process
- Switching performance is guaranteed over full temperature and V_{CC} supply range
- Pin and functional compatible with LS family counterpart
- Improved output transient handling capability

Connection Diagram

Dual-In-Line Package



TL/F/6204–1

Order Number DM74ALS153M, DM74ALS153N or DM74ALS153SJ
See NS Package Number M16A, M16D or N16A

Function Table

Select Inputs		Data Inputs				Strobe	Output
B	A	C0	C1	C2	C3	G	Y
X	X	X	X	X	X	H	L
L	L	L	X	X	X	L	L
L	L	H	X	X	X	L	H
L	H	X	L	X	X	L	L
L	H	X	H	X	X	L	H
H	L	X	X	L	X	L	L
H	L	X	X	H	X	L	H
H	H	X	X	X	L	L	L
H	H	X	X	X	H	L	H

Select inputs A and B are common to both sections.

H = High Level, L = Low Level, X = Don't Care

©1995 National Semiconductor Corporation

RRD-B30M105/Printed in U. S. A.

Absolute Maximum Ratings

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
DM74ALS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C
Typical θ_{JA}	
N Package	78.0°C/W
M Package	107.0°C/W

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter	DM74ALS153			Units
		Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	V
V_{IH}	High Level Input Voltage	2			V
V_{IL}	Low Level Input Voltage			0.8	V
I_{OH}	High Level Output Current			−2.6	mA
I_{OL}	Low Level Output Current			24	mA
T_A	Free Air Operating Temperature	0		70	°C

Electrical Characteristics

over recommended operating free air temperature range. All typical values are measured at V_{CC} = 5V, T_A = 25°C.

Symbol	Parameter	Conditions		Min	Typ	Max	Units
V_{IK}	Input Clamp Voltage	V_{CC} = 4.5V, I_{IN} = −18 mA				−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = 4.5V, I_{OH} = Max		2.4	3.2		V
		I_{OH} = −400 μA, V_{CC} = 4.5V to 5.5V		V_{CC} − 2			V
V_{OL}	Low Level Output Voltage	V_{CC} = 4.5V	74ALS I_{OL} = 24 mA		0.35	0.5	V
I_I	Input Current at Max Input Voltage	V_{CC} = 5.5V, V_{IN} = 7V				0.1	mA
I_{IH}	High Level Input Current	V_{CC} = 5.5V, V_{IN} = 2.7V				20	μA
I_{IL}	Low Level Input Current	V_{CC} = 5.5V, V_{IN} = 0.4V				−0.1	mA
I_O	Output Drive Current	V_{CC} = 5.5V, V_{OUT} = 2.25V		−30		−112	mA
I_{CC}	Supply Current	V_{CC} = 5.5V All Inputs = 4.5V			7.5	14	mA

National Semiconductor

May 1989

DM54LS154/DM74LS154 4-Line to 16-Line Decoders/Demultiplexers

General Description

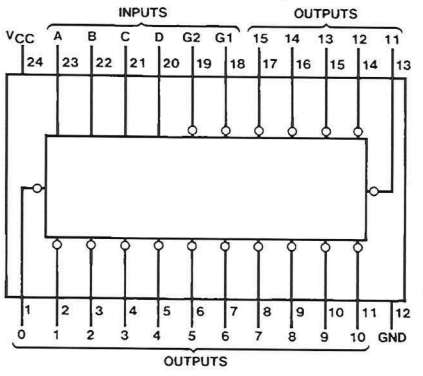
Each of these 4-line-to-16-line decoders utilizes TTL circuitry to decode four binary-coded inputs into one of sixteen mutually exclusive outputs when both the strobe inputs, G1 and G2, are low. The demultiplexing function is performed by using the 4 input lines to address the output line, passing data from one of the strobe inputs with the other strobe input low. When either strobe input is high, all outputs are high. These demultiplexers are ideally suited for implementing high-performance memory decoders. All inputs are buffered and input clamping diodes are provided to minimize transmission-line effects and thereby simplify system design.

Features

- Decodes 4 binary-coded inputs into one of 16 mutually exclusive outputs
- Performs the demultiplexing function by distributing data from one input line to any one of 16 outputs
- Input clamping diodes simplify system design
- High fan-out, low-impedance, totem-pole outputs
- Typical propagation delay
  3 levels of logic 23 ns
  Strobe 19 ns
- Typical power dissipation 45 mW

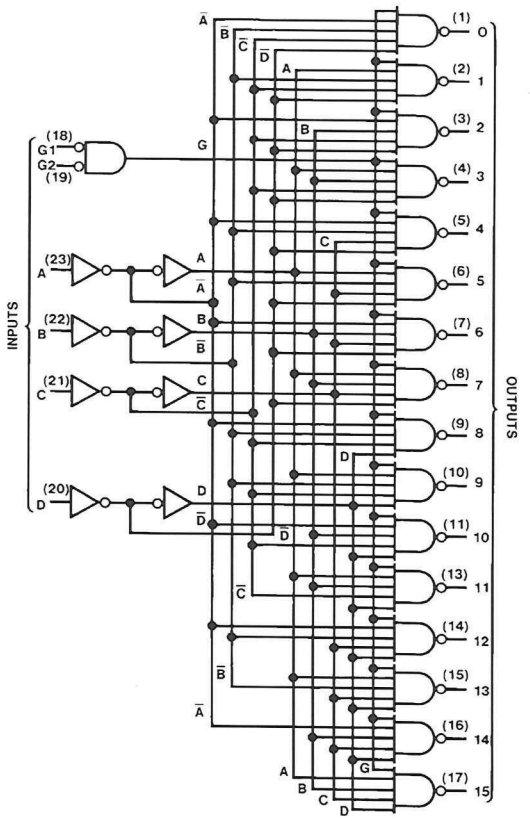
Connection and Logic Diagrams

Dual-In-Line Package



TL/F/6394–1

Order Number DM54LS154J,
DM74LS154WM or DM74LS154N
See NS Package Number J24A, M24B or N24A



TL/F/6394–2

©1995 National Semiconductor Corporation

RRD-B30M105/Printed in U. S. A.

Function Table

Inputs						Outputs													
G1	G2	D	C	B	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	L	L	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
L	L	H	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
L	L	H	H	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H
L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L
L	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	L	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H

H = High Level, L = Low Level, X = Don't Care

National Semiconductor

June 1989

54LS164/DM54LS164/DM74LS164
8-Bit Serial In/Parallel Out Shift Registers

General Description

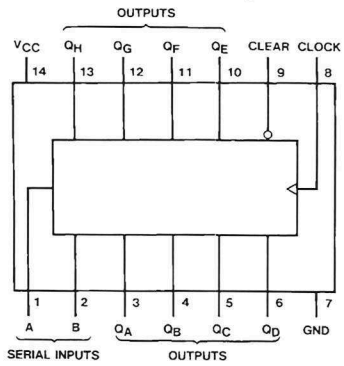
These 8-bit shift registers feature gated serial inputs and an asynchronous clear. A low logic level at either input inhibits entry of the new data, and resets the first flip-flop to the low level at the next clock pulse, thus providing complete control over incoming data. A high logic level on either input enables the other input, which will then determine the state of the first flip-flop. Data at the serial inputs may be changed while the clock is high or low, but only information meeting the setup and hold time requirements will be entered. Clocking occurs on the low-to-high level transition of the clock input. All inputs are diode-clamped to minimize transmission-line effects.

Features

- Gated (enable/disable) serial inputs
- Fully buffered clock and serial inputs
- Asynchronous clear
- Typical clock frequency 36 MHz
- Typical power dissipation 80 mW
- Alternate Military/Aerospace device (54LS164) is available. Contact a National Semiconductor Sales Office/ Distributor for specifications.

Connection Diagram

Dual-In-Line Package



TL/F/6398–1

Order Number 54LS164DMQB, 54LS164FMQB, 54LS164LMQB, DM54LS164J, DM54LS164W, DM74LS164M or DM74LS164N
See NS Package Number E20A, J14A, M14A, N14A or W14B

Function Table

Inputs				Outputs			
Clear	Clock	A	B	Q_A	Q_B	...	Q_H
L	X	X	X	L	L	...	L
H	L	X	X	Q_{A0}	Q_{B0}	...	Q_{H0}
H	↑	H	H	H	Q_{An}	...	Q_{Gn}
H	↑	L	X	L	Q_{An}	...	Q_{Gn}
H	↑	X	L	L	Q_{An}	...	Q_{Gn}

H = High Level (steady state), L = Low Level (steady state)

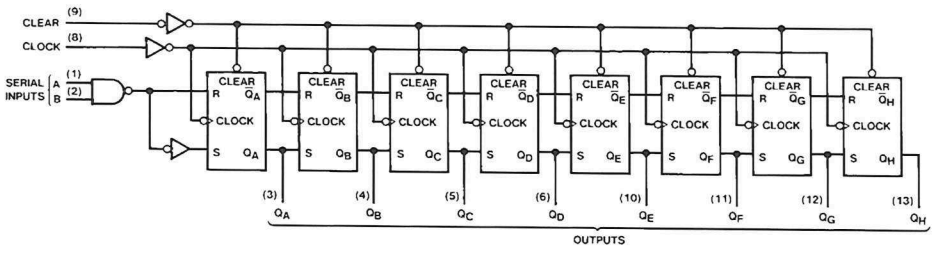
X = Don't Care (any input, including transitions)

↑ = Transition from low to high level

Q_{A0}, Q_{B0}, Q_{H0} = The level of Q_A, Q_B, or Q_H, respectively, before the indicated steady-state input conditions were established.

Q_{An}, Q_{Gn} = The level of Q_A or Q_G before the most recent ↑ transition of the clock; indicates a one-bit shift.

Logic Diagram



TL/F/6398–2

©1995 National Semiconductor Corporation

RRD-B30M105/Printed in U. S. A.

Absolute Maximum Ratings (Note)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
DM54LS and 54LS	−55°C to +125°C
DM74LS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" tables will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter		DM54LS164			DM74LS164			Units
			Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage		4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High Level Input Voltage		2			2			V
V_{IL}	Low Level Input Voltage				0.7			0.8	V
I_{OH}	High Level Output Current				−0.4			−0.4	mA
I_{OL}	Low Level Output Current				4			8	mA
f_{CLK}	Clock Frequency (Note 4)		0		25	0		25	MHz
t_W	Pulse Width (Note 4)	Clock	20			20			ns
		Clear	20			20			
t_{SU}	Data Setup Time (Note 4)		17			17			ns
t_H	Data Hold Time (Note 4)		5			5			ns
t_{REL}	Clear Release Time (Note 4)		30			30			ns
T_A	Free Air Operating Temperature		−55		125	0		70	°C

Electrical Characteristics over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions		Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA				−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max, V_{IH} = Min	DM54	2.5	3.4		V
			DM74	2.7	3.4		
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IL} = Max, V_{IH} = Min	DM54		0.25	0.4	V
			DM74		0.35	0.5	
		I_{OL} = 4 mA, V_{CC} = Min	DM74		0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V				0.1	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V				20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V				−0.4	mA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 2)	DM54	−20		−100	mA
			DM74	−20		−100	
I_{CC}	Supply Current	V_{CC} = Max (Note 3)			16	27	mA

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Note 3: I_{CC} is measured with all outputs open, the SERIAL input grounded, the CLOCK input at 2.4V, and a momentary ground, then 4.5V, applied to the CLEAR input.

Note 4: T_A = 25°C and V_{CC} = 5V.

National Semiconductor

PRELIMINARY
May 1987

DM54ALS192/DM74ALS192/DM54ALS193/DM74ALS193 Synchronous Four-Bit Up/Down Counters (Dual Clock with Clear)

DM54ALS192/DM74ALS192/DM54ALS193/DM74ALS193
Synchronous Four-Bit Up/Down Counters (Dual Clock with Clear)

General Description

The 'ALS192 and 'ALS193 are synchronous, reversible up/down counters. The 'ALS192 is a 4-bit decade counter and the 'ALS193 is a 4-bit binary counter. Synchronous operation is provided by having all flip-flops clocked simultaneously so that the outputs change coincident with each other when so instructed by the steering logic. This mode of operation eliminates the output counting spikes normally associated with asynchronous (ripple clock) counters.

The outputs of the four flip-flops are triggered by a low-to-high level transition of either count (clock) input (up or down). The direction of counting is determined by which count input is pulsed while the other count input is high.

All four counters are fully programmable; that is, each output may be preset to either level by placing a low on the load input and entering the desired data at the data inputs. The output will change to agree with the data inputs independently of the count pulses. This feature allows the counters to be used as modulo-N dividers by simply modifying the count length with the preset inputs.

A clear input has been provided that forces all outputs to the low level when a high level is applied. The clear function is independent of the count and the load inputs. The clock, count, and load inputs are buffered to lower the drive requirements. This significantly reduces the loading on clock drivers, etc., for long parallel words.

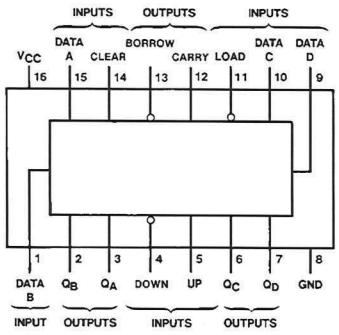
These counters were designed to be cascaded without the need for external circuitry. The borrow output (BO) produces a low-level pulse while the count is zero (all outputs low) and the count-down input is low. Similarly, the carry output ($\overline{CO}$) produces a low-level pulse while the count is maximum (9 or 15) and the count-up input is low. The counters can then be easily cascaded by feeding the borrow and carry outputs to the count-down and count-up inputs, respectively, of the succeeding counter.

Features

- Switching specifications at 50 pF
- Switching specifications guaranteed over full temperature and V_{CC} range
- Advanced oxide-isolated, ion-implanted Schottky TTL process
- Functionally and pin-for-pin compatible with Schottky and low power Schottky TTL counterparts
- Improved AC performance over Schottky and low power Schottky counterparts
- Look ahead circuitry enhances cascaded counters
- Fully synchronous in count mode
- Parallel asynchronous load for modulo-N count lengths
- Asynchronous clear

Connection Diagram

Dual-In-Line Package



TL/F/6209–1

Order Number DM54ALS192J, 193J, DM74ALS192M, 193M or DM74ALS192N, 193N
See NS Package Number J16A, M16A or N16A

This document contains information on a device under development. National Semiconductor Corporation reserves the right to change or discontinue this product without notice.

©1995 National Semiconductor Corporation TL/F/6209

RRD-B30M75/Printed in U. S. A.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
DM54ALS	−55°C to +125°C
DM74ALS	0 to +70°C
Storage Temperature Range	−65°C to +150°C

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

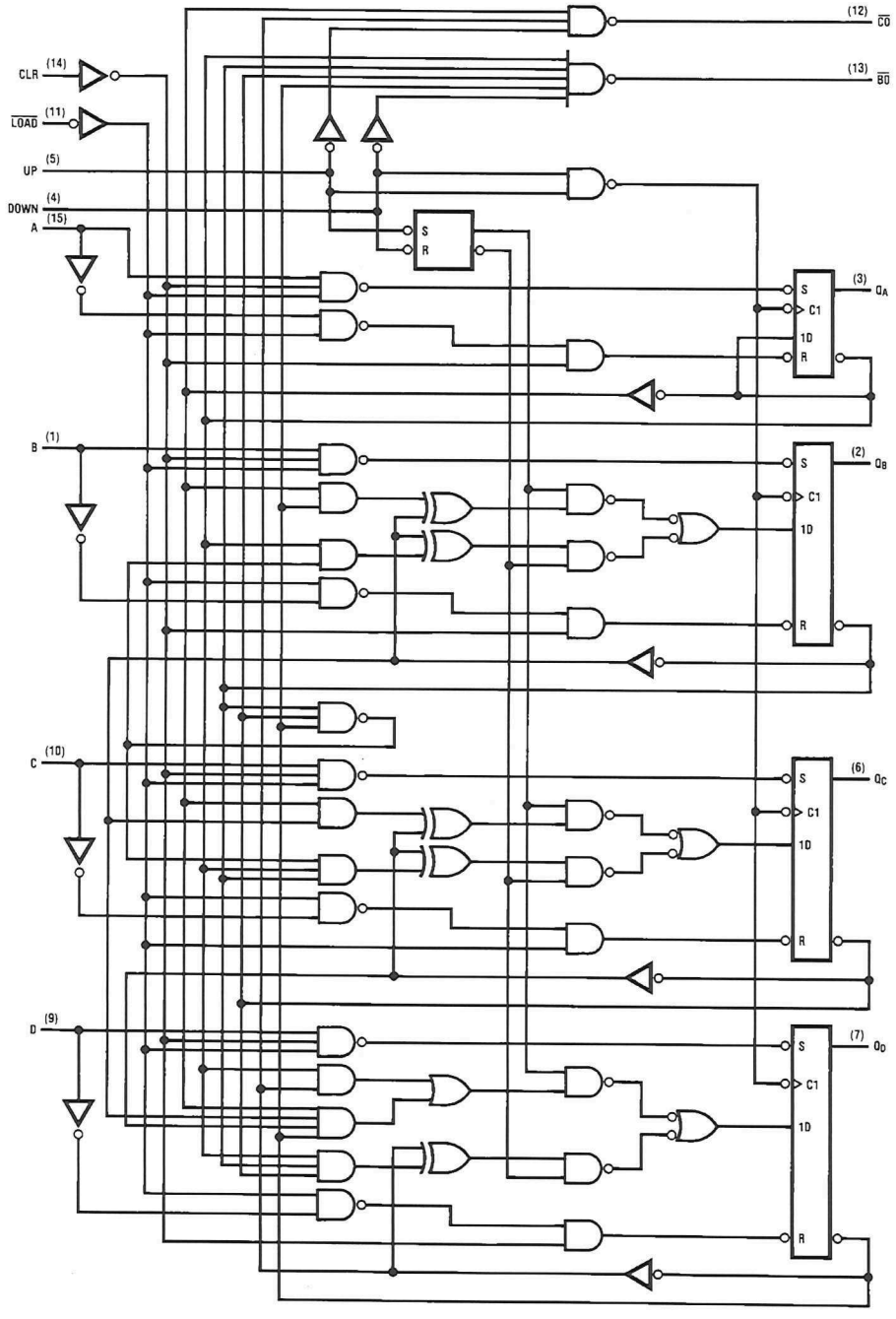
Symbol	Parameter			DM54ALS192, 193			DM74ALS192, 193			Units
				Min	Typ	Max	Min	Typ	Max	
V_{CC}	Supply Voltage			4.5	5	5.5	4.5	5	5.5	V
V_{IH}	High Level Input Voltage			2			2			V
V_{IL}	Low Level Input Voltage					0.7			0.8	V
I_{OH}	High Level Output Current					−0.4			−0.4	mA
I_{OL}	Low Level Output Current					4			8	mA
f_{CLK}	Clock Frequency	'ALS192		0		20	0		25	MHz
		'ALS193		0		25	0		30	MHz
t_W	Pulse Duration	CLR High		10			10			ns
		$\overline{LOAD}$ Low		25			20			ns
		UP or DOWN High or Low	'ALS192	25			20			ns
			'ALS193	30			16.5			ns
t_{SU}	Setup Time	Data before $\overline{LOAD}$ ↑		25			20			ns
		CLR Inactive before UP ↑ or DOWN ↑		20			20			ns
		$\overline{LOAD}$ Inactive before UP ↑ or DOWN ↑		20			20			ns
		UP High before DOWN ↑		17			17			ns
		DOWN High before UP ↑		15			15			ns
t_H	Hold Time	Data after $\overline{LOAD}$ ↑		5			5			ns
		UP High after DOWN ↑		0			0			ns
		DOWN High after UP ↑		0			0			ns
T_A	Operating Free Air Temperature			−55		125	0		70	°C

Electrical Characteristics over recommended operating free air temperature range

Symbol	Parameter	Conditions			DM74ALS192, 193			Units
					Min	Typ (Note 1)	Max	
V_{IK}	Input Clamp Voltage	V_{CC} = 4.5V, I_I = −18 mA					−1.5	V
V_{OH}	High Level Output	V_{CC} = 4.5V to 5.5V, I_{OH} = −0.4 mA			V_{CC} − 2			V
V_{OL}	Low Level Output Voltage	V_{CC} = 4.5V	54/74ALS I_{OL} = 4 mA			0.25	0.4	V
		V_{CC} = 4.5V	74ALS I_{OL} = 8 mA			0.35	0.5	V
I_I	Input Current at Max Input Voltage	V_{CC} = 5.5V, V_I = 7V					0.1	mA
I_{IH}	High Level Input Current	V_{CC} = 5.5V, V_I = 2.7V					20	μA
I_{IL}	Low Level Input Current	V_{CC} = 5.5V, V_I = 0.4V		UP, DOWN			−0.2	mA
				All Others			−0.1	mA

Logic Diagrams

'ALS192

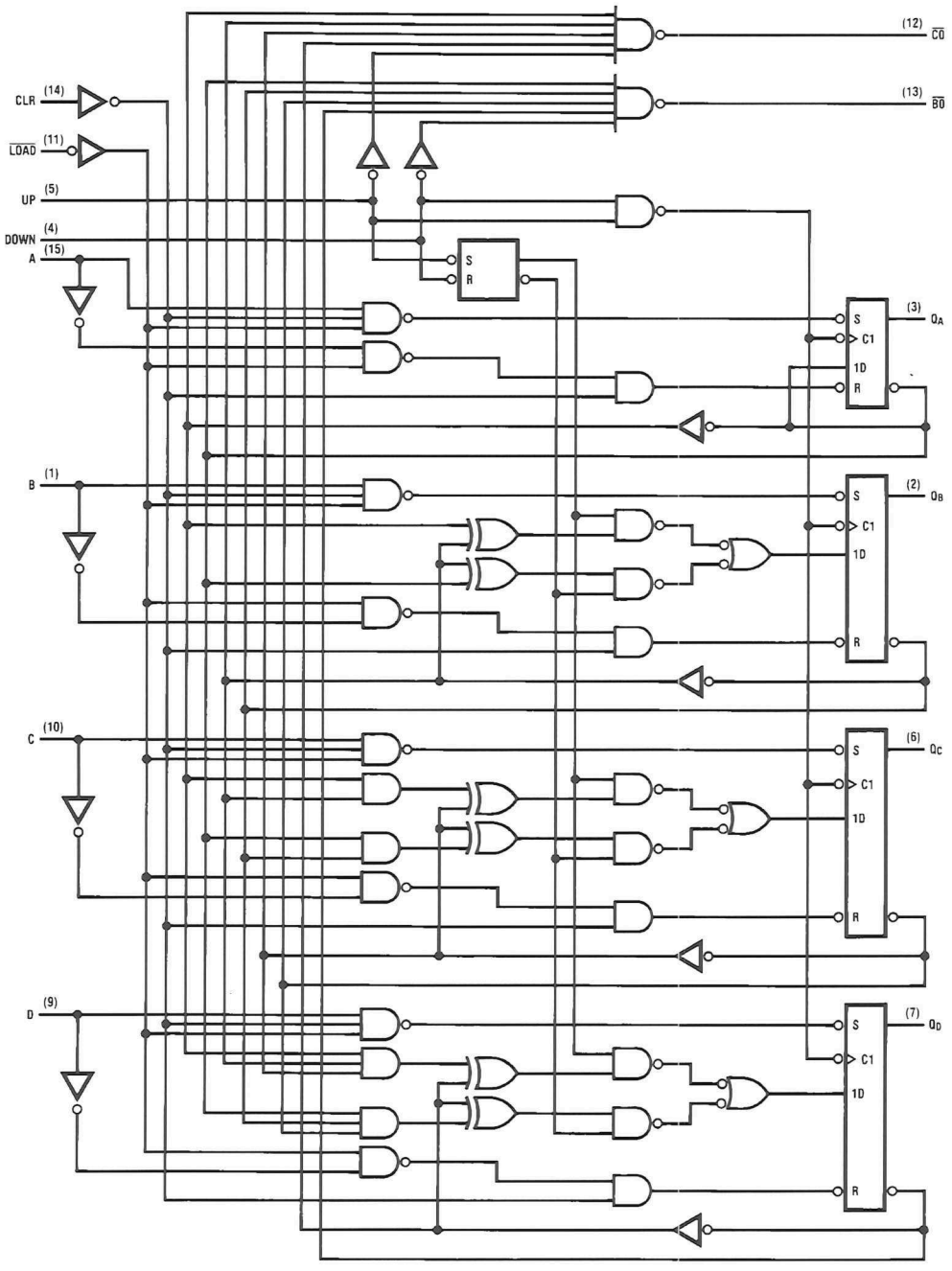


TL/F/6209–2

Pin numbers shown are for J and N packages.

Logic Diagrams (Continued)

'ALS193



TL/F/6209–3

Pin numbers shown are for J and N packages.

National Semiconductor

June 1989

54LS194A/DM74LS194A 4-Bit Bidirectional Universal Shift Register

General Description

This bidirectional shift register is designed to incorporate virtually all of the features a system designer may want in a shift register; they feature parallel inputs, parallel outputs, right-shift and left-shift serial inputs, operating-mode-control inputs, and a direct overriding clear line. The register has four distinct modes of operation, namely:

Parallel (broadside) load
Shift right (in the direction Q_A toward Q_D)
Shift left (in the direction Q_D toward Q_A)
Inhibit clock (do nothing)

Synchronous parallel loading is accomplished by applying the four bits of data and taking both mode control inputs, S0 and S1, high. The data is loaded into the associated flip-flops and appear at the outputs after the positive transition of the clock input. During loading, serial data flow is inhibited.

Shift right is accomplished synchronously with the rising edge of the clock pulse when S0 is high and S1 is low. Serial data for this mode is entered at the shift-right data input. When S0 is low and S1 is high, data shifts left synchronously and new data is entered at the shift-left serial input.

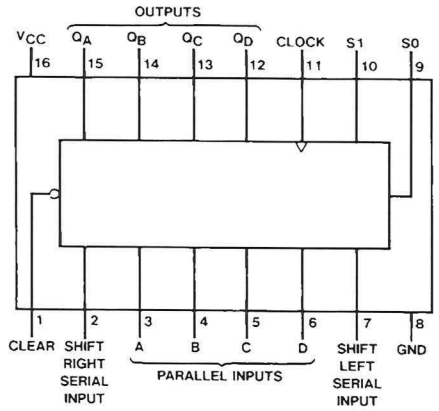
Clocking of the flip-flop is inhibited when both mode control inputs are low.

Features

- Parallel inputs and outputs
- Four operating modes:
  - Synchronous parallel load
  - Right shift
  - Left shift
  - Do nothing
- Positive edge-triggered clocking
- Direct overriding clear

Connection Diagram

Dual-In-Line Package



TL/F/6407–1

Order Number 54LS194ADMQB, 54LS194AFMQB, 54LS194ALMQB, DM74LS194AM or DM74LS194AN
See NS Package Number E20A, J16A, M16A, N16E or W16A

©1995 National Semiconductor Corporation

RRD-B30M105/Printed in U. S. A.

Switching Characteristics at V_{CC} = 5V and T_A = 25°C (See Section 1 for Test Waveforms and Output Load)

Symbol	Parameter	From (Input) To (Output)	54LS		DM74LS		Units
			C_L = 15 pF		C_L = 50 pF, R_L = 2 kΩ		
			Min	Max	Min	Max	
f_{MAX}	Maximum Clock Frequency		30		20		MHz
t_{PLH}	Propagation Delay Time Low to High Level Output	Clock to Any Q		21		26	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Clock to Any Q		24		35	ns
t_{PHL}	Propagation Delay Time High to Low Output	Clear to Any Q		26		38	ns

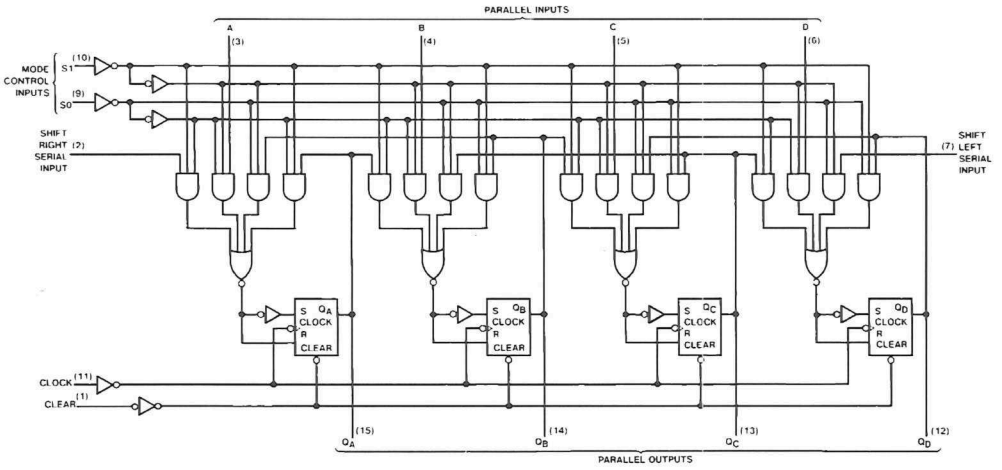
Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Note 3: With all outputs open, inputs A through D grounded, and 4.5V applied to S0, S1, CLEAR, and the serial inputs, I_{CC} is tested with momentary ground, then 4.5V applied to CLOCK.

Logic Diagram

LS194A



TL/F/6407–2

Function Table

Inputs										Outputs			
Clear	Mode		Clock	Serial		Parallel				Q_A	Q_B	Q_C	Q_D
	S1	S0		Left	Right	A	B	C	D				
L	X	X	X	X	X	X	X	X	X	L	L	L	L
H	X	X	L	X	X	X	X	X	X	Q_{A0}	Q_{B0}	Q_{C0}	Q_{D0}
H	H	H	↑	X	X	a	b	c	d	a	b	c	d
H	L	H	↑	X	H	X	X	X	X	H	Q_{An}	Q_{Bn}	Q_{Cn}
H	L	H	↑	X	L	X	X	X	X	L	Q_{An}	Q_{Bn}	Q_{Cn}
H	H	L	↑	H	X	X	X	X	X	Q_{Bn}	Q_{Cn}	Q_{Dn}	H
H	H	L	↑	L	X	X	X	X	X	Q_{Bn}	Q_{Cn}	Q_{Dn}	L
H	L	L	X	X	X	X	X	X	X	Q_{A0}	Q_{B0}	Q_{C0}	Q_{D0}

H = High Level (steady state), L = Low Level (steady state), X = Don't Care (any input, including transitions)

↑ = Transition from low to high level

a, b, c, d = The level of steady state input at inputs A, B, C or D, respectively.

Q_{A0}, Q_{B0}, Q_{C0}, Q_{D0} = The level of Q_A, Q_B, Q_C, or Q_D, respectively, before the indicated steady state input conditions were established.

Q_{An}, Q_{Bn}, Q_{Cn}, Q_{Dn} = The level of Q_A, Q_B, Q_C, respectively, before the most-recent ↑ transition of the clock.

National Semiconductor

February 1992

DM74LS221 Dual Non-Retriggerable One-Shot with Clear and Complementary Outputs

General Description

The DM74LS221 is a dual monostable multivibrator with Schmitt-trigger input. Each device has three inputs permitting the choice of either leading-edge or trailing-edge triggering. Pin (A) is an active-low trigger transition input and pin (B) is an active-high transition Schmitt-trigger input that allows jitter free triggering for inputs with transition rates as slow as 1 volt/second. This provides the input with excellent noise immunity. Additionally an internal latching circuit at the input stage also provides a high immunity to V_{CC} noise. The clear (CLR) input can terminate the output pulse at a predetermined time independent of the timing components. This (CLR) input also serves as a trigger input when it is pulsed with a low level pulse transition (⌐⌙). To obtain the best and trouble free operation from this device please read operating rules as well as the NSC one-shot application notes carefully and observe recommendations.

Features

- A dual, highly stable one-shot
- Compensated for V_{CC} and temperature variations
- Pin-out identical to 'LS123 (Note 1)
- Output pulse width range from 30 ns to 70 seconds
- Hysteresis provided at (B) input for added noise immunity
- Direct reset terminates output pulse
- Triggerable from CLEAR input
- DTL, TTL compatible
- Input clamp diodes

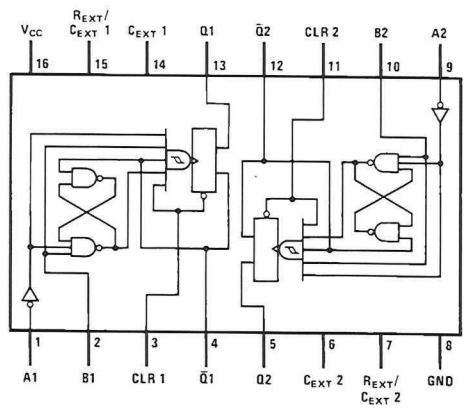
Note 1: The pin-out is identical to 'LS123 but, functionally it is not; refer to Operating Rules #10 in this datasheet.

Functional Description

The basic output pulse width is determined by selection of an external resistor (R_X) and capacitor (C_X). Once triggered, the basic pulse width is independent of further input transitions and is a function of the timing components, or it may be reduced or terminated by use of the active low CLEAR input. Stable output pulse width ranging from 30 ns to 70 seconds is readily obtainable.

Connection Diagram

Dual-In-Line Package



TL/F/6409–1

Order Number DM74LS221M or DM74LS221N
See NS Package Number M16A or N16A

Function Table

Inputs			Outputs	
CLEAR	A	B	Q	$\overline{Q}$
L	X	X	L	H
X	H	X	L	H
X	X	L	L	H
H	L	↑	⎍	⎌
H	↓	H	⎍	⎌
* ↑	L	H	⎍	⎌

H = High Logic Level

L = Low Logic Level

X = Can Be Either Low or High

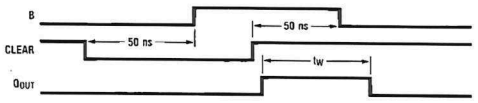
↑ = Positive Going Transition

↓ = Negative Going Transition

⎍ = A Positive Pulse

⎌ = A Negative Pulse

*This mode of triggering requires first the B input be set from a low to high level while the CLEAR input is maintained at logic low level. Then with the B input at logic high level, the CLEAR input whose positive transition from low to high will trigger an output pulse.



TL/F/6409–2

©1995 National Semiconductor Corporation TL/F/6409

RRD-B30M105/Printed in U. S. A.

DM74LS221 Dual Non-Retriggerable One-Shot with Clear and Complementary Outputs

Absolute Maximum Ratings (Note)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
DM74LS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter		DM74LS221			Units
			Min	Nom	Max	
V_{CC}	Supply Voltage		4.75	5	5.25	V
V_{T+}	Positive-Going Input Threshold Voltage at the A Input (V_{CC} = Min)			1	2	V
V_{T-}	Negative-Going Input Threshold Voltage at the A Input (V_{CC} = Min)		0.8	1		V
V_{T+}	Positive-Going Input Threshold Voltage at the B Input (V_{CC} = Min)			1	2	V
V_{T-}	Negative-Going Input Threshold Voltage at the B Input (V_{CC} = Min)		0.8	0.9		V
I_{OH}	High Level Output Current				−0.4	mA
I_{OL}	Low Level Output Current				8	mA
t_W	Pulse Width (Note 1)	Data	40			ns
		Clear	40			
t_{REL}	Clear Release Time (Note 1)		15			ns
$\frac{dV}{dt}$	Rate of Rise or Fall of Schmitt Input (B) (Note 1)				1	$\frac{V}{s}$
$\frac{dV}{dt}$	Rate of Rise or Fall of Logic Input (A) (Note 1)				1	$\frac{V}{\mu s}$
R_{EXT}	External Timing Resistor (Note 1)		1.4		100	kΩ
C_{EXT}	External Timing Capacitance (Note 1)		0		1000	μF
DC	Duty Cycle (Note 1)	R_T = 2 kΩ			50	%
		R_T = R_{EXT} (Max)			60	
T_A	Free Air Operating Temperature		0		70	°C

Note 1: T_A = 25°C and V_{CC} = 5V.

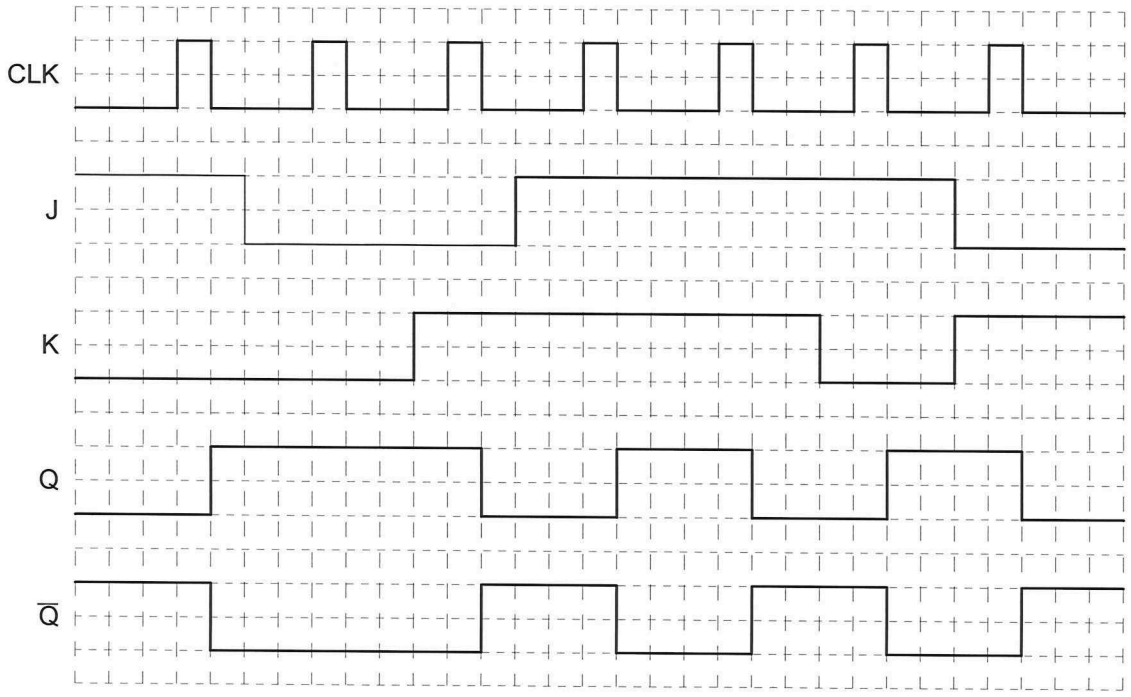
Electrical Characteristics over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA			−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max, V_{IH} = Min	2.7	3.4		V
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IL} = Max, V_{IH} = Min		0.35	0.5	V
		V_{CC} = Min, I_{OL} = 4 mA			0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V			0.1	mA

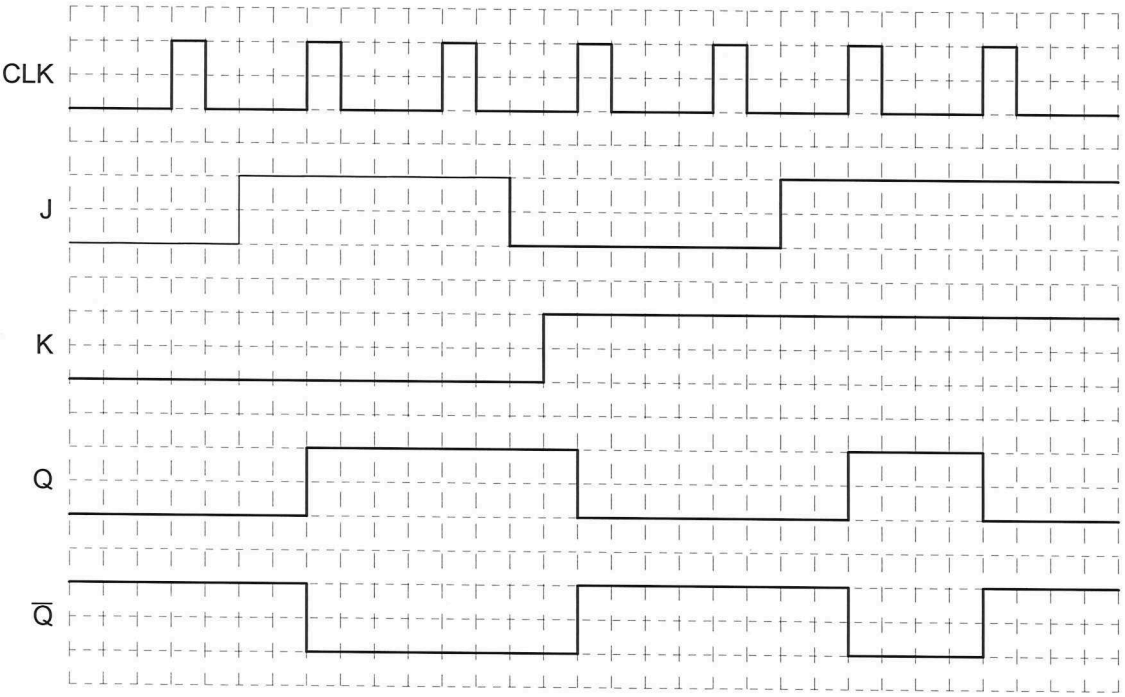
Answers to review questions

Section 1

1.



2.



Copyright: Not to be reproduced without permission

Section 1 (continued)

3. *P* and *C* only

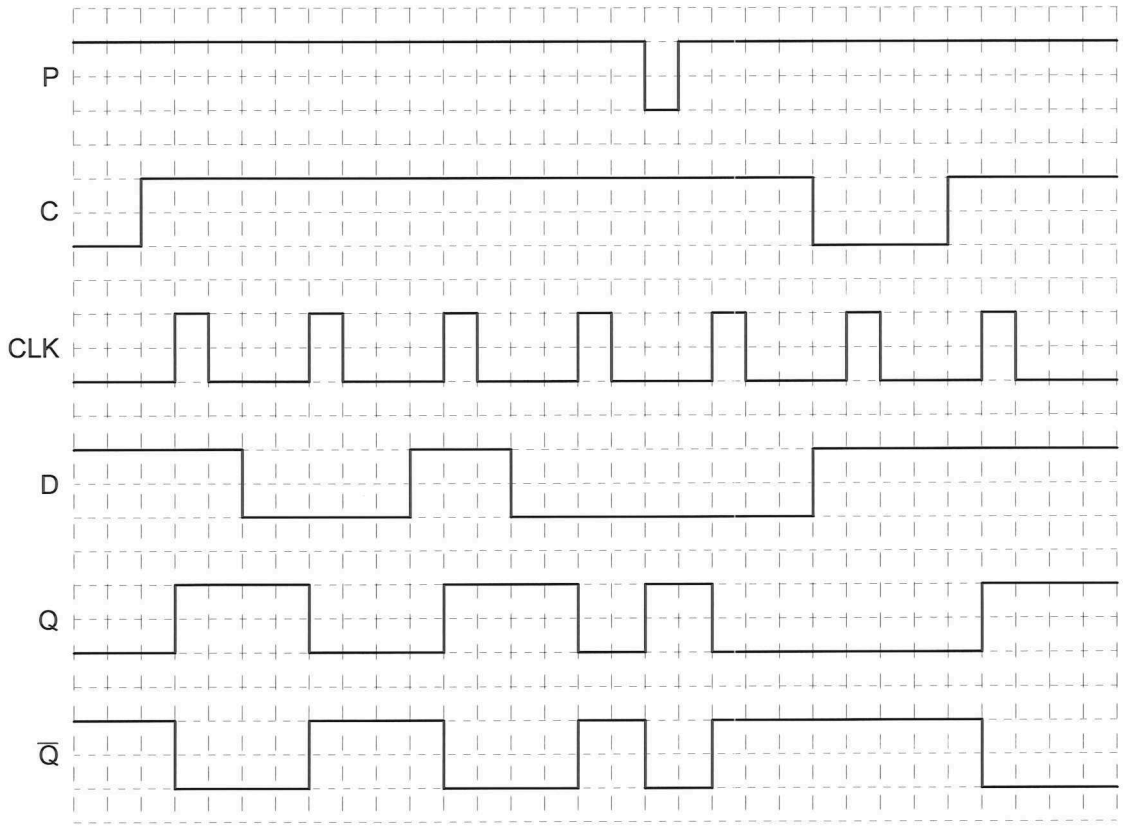
4. *D* only

5. None of them

6. *P* and *C* only

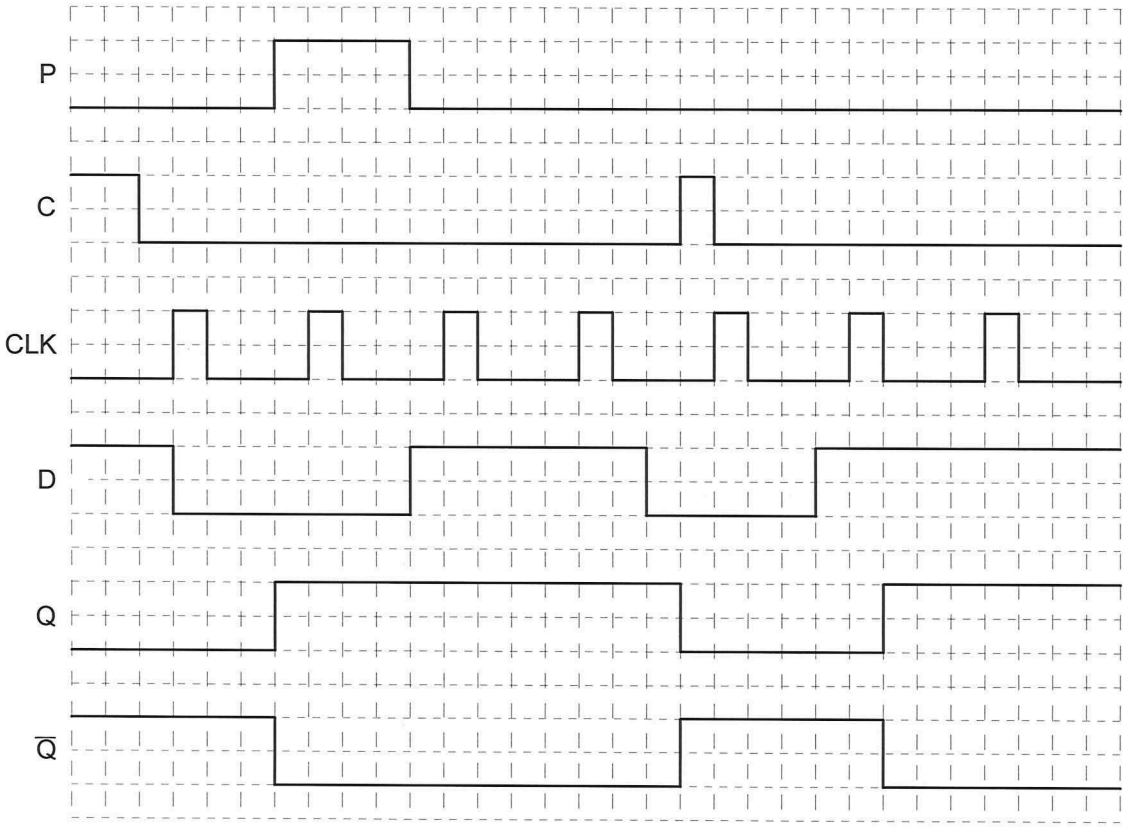
7. *CLK* only

8.



Section 1 (continued)

9.



10. A latch
A NOR or NAND RS FF
A frequency divider
A counter
A shift register

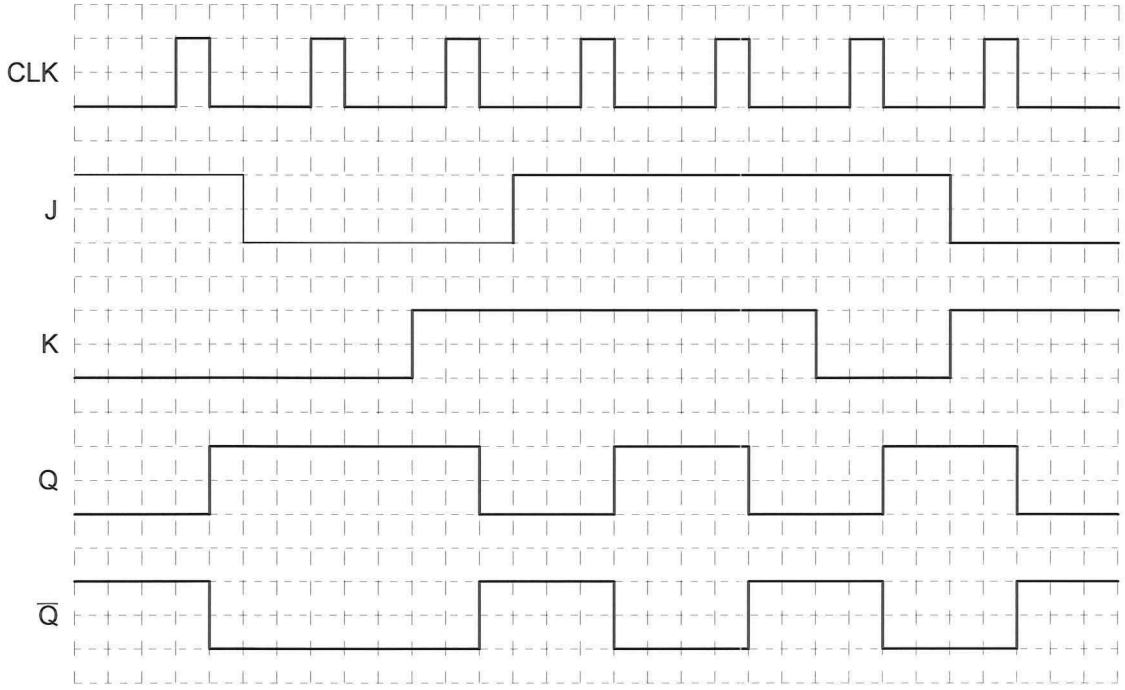
11. Frequency divider

12. The output signal is half the frequency of the input signal (or twice the period).

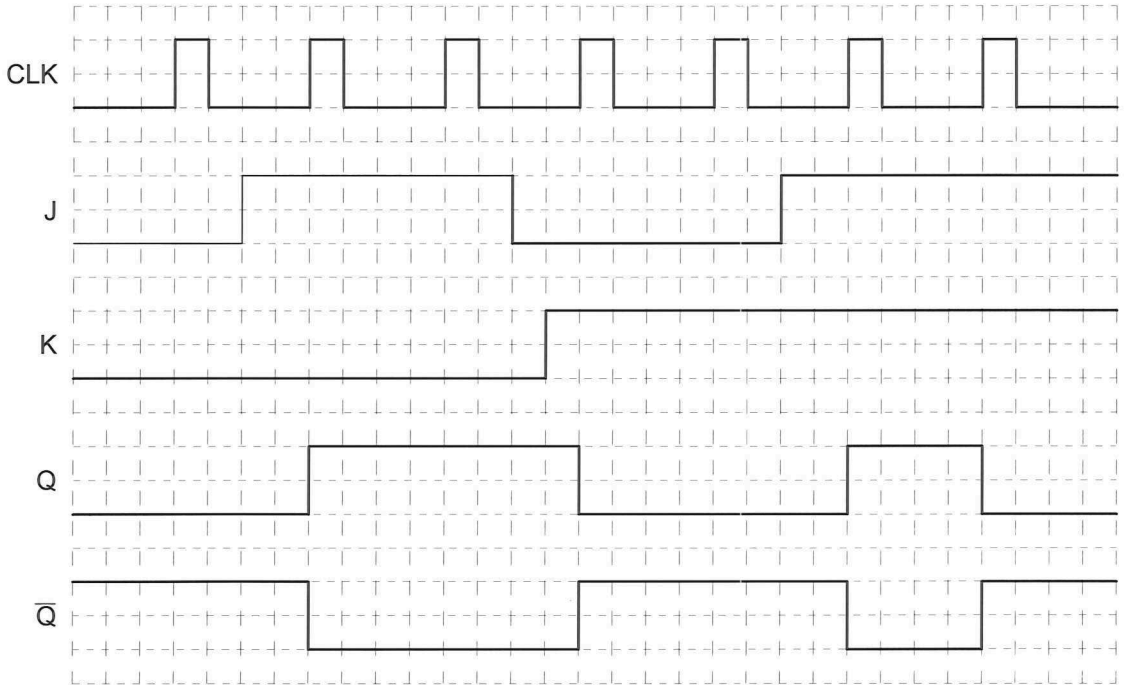
Copyright: Not to be reproduced without permission

Section 2

1.



2.



Section 2 (continued)

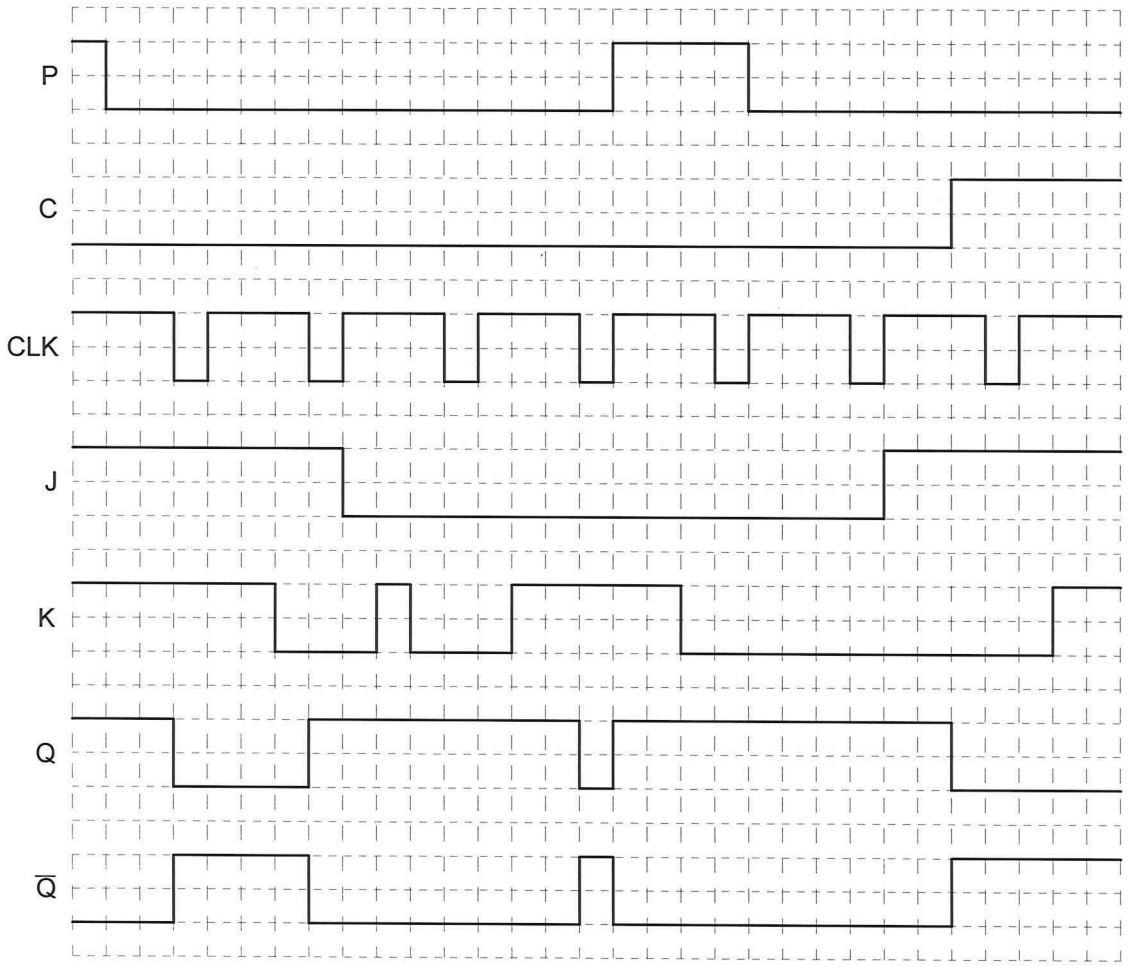
3. *P* and *C* only

4. *J* and *K* only

5. *CLK* only

6. None of them

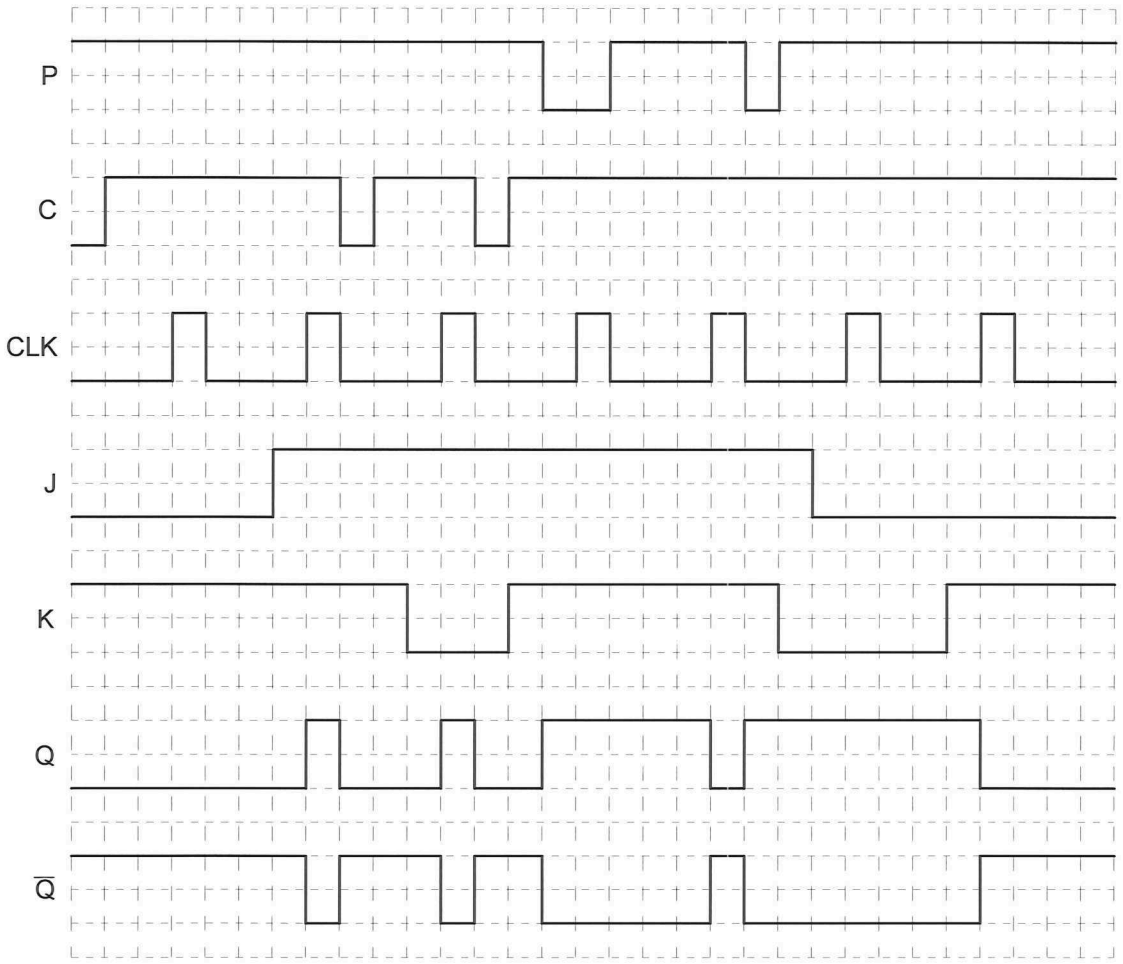
7.



Copyright: Not to be reproduced without permission

Section 2 (continued)

8.



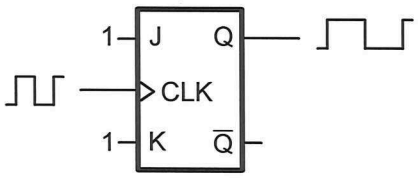
9. hard-wired to logic-1. [Note: If the preset and clear inputs are active high instead of active low then they should be hard-wired to logic-0.]

10. The voltage on these inputs floats which can cause the preset and clear inputs to be randomly asserted which can disrupt the operation of the circuit.

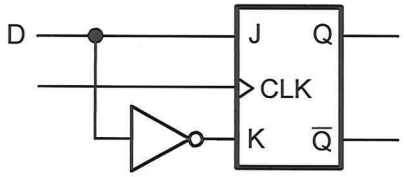
11. A latch
A NOR or NAND RS FF
The J-K FF can be wired to operate as a D-type FF (but not the other way around)
A frequency divider
A counter
A shift register

Section 2 (continued)

12.



13.



14. 7473

15. 4027

16. 7473 and 7476

17. 4027

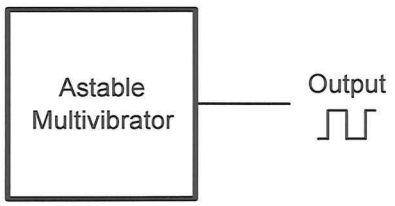
18. None of them

19. No! [Also, notice that supply pins on the 7476 are not in the usual place.]

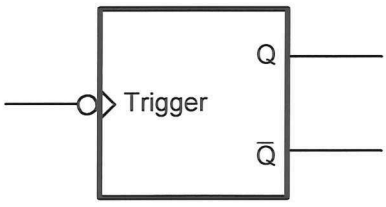
 Copyright: Not to be reproduced without permission

Section 3

1. two logic levels.
2. either continuously or on demand depending on the device.
3. two stable states.
4. A NAND RS FF, a NOR RS FF and a J-K FF
5. no stable states.
6. A squarewave oscillator
7. one stable state.
8. A oneshot
9. Astable
10. Monostable
11.



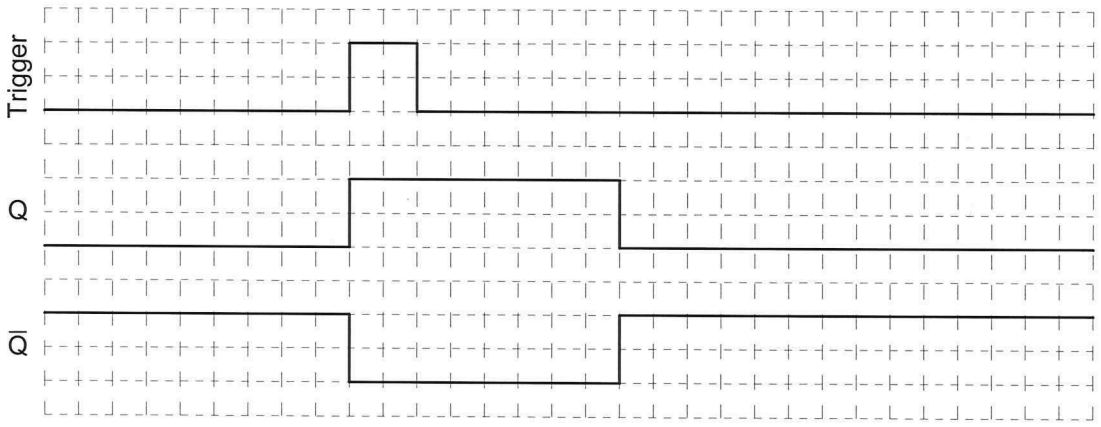
12. All microprocessor-based digital electronics equipment.
13.



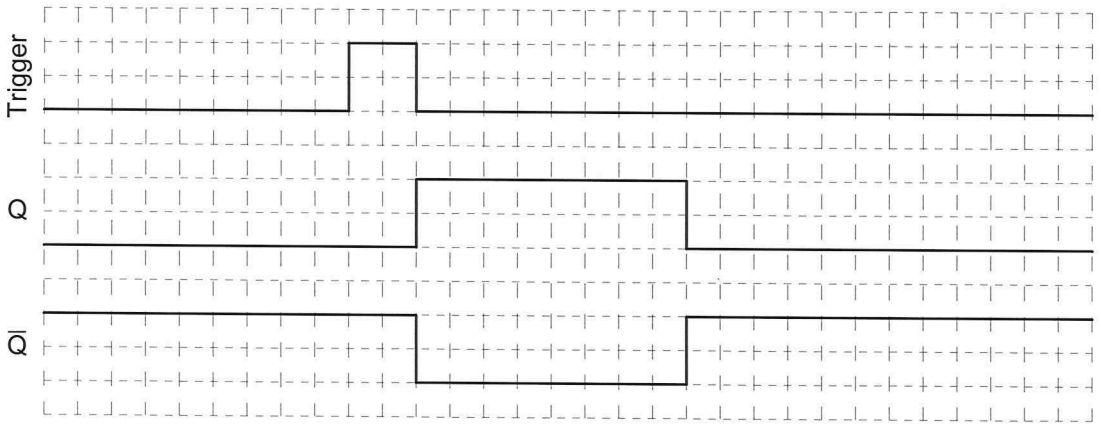
Section 3 (continued)

14. Access control systems (like boom gates for car parks)
Pulse stretching

15.



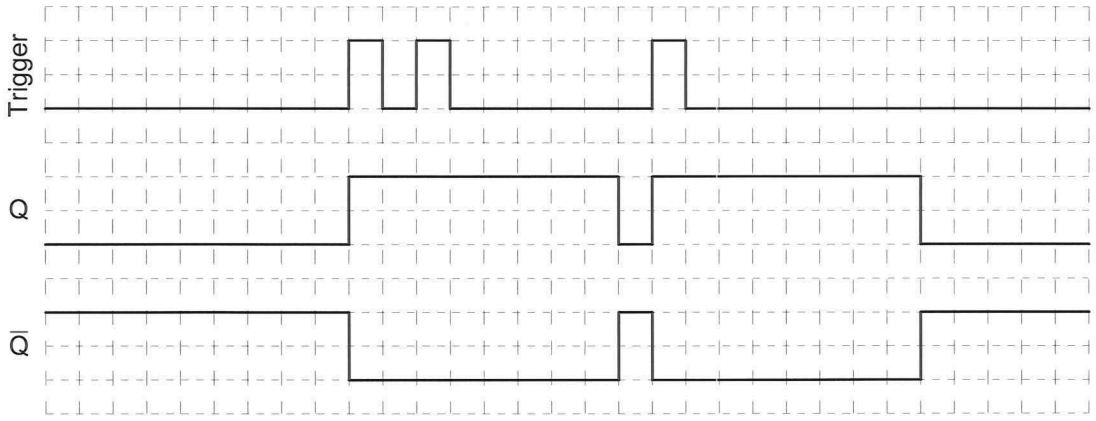
16.



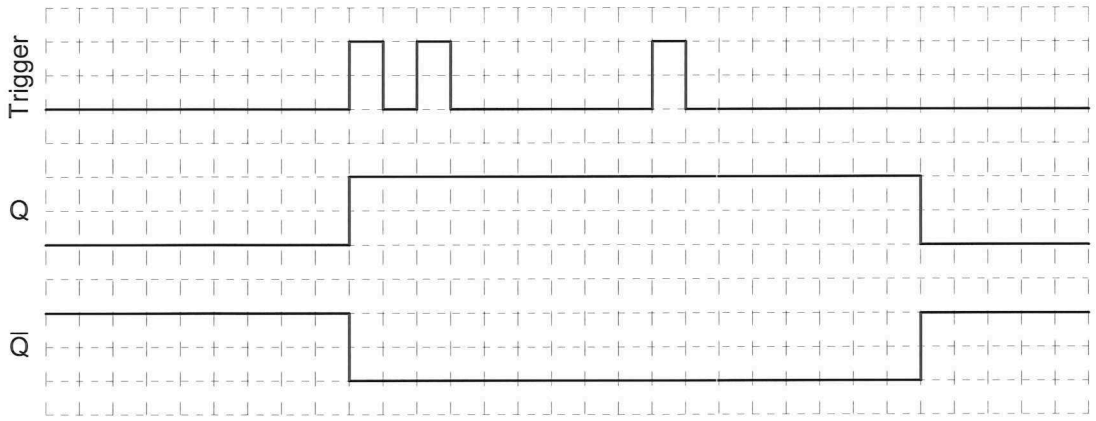
 Copyright: Not to be reproduced without permission

Section 3 (continued)

17.



18.



19. retriggerable.

20. two trigger inputs - one positive and one negative edge-triggered.

21. active low.

22. 5kΩ to 260kΩ [Have a good look, the information is in the data!]

23. There are no restrictions on the value of capacitance used.

Section 3 (continued)

24. 116ns

25. $t_w = 0.45 \times R_{ext} \times C_{ext}$

26. $t_w = 0.45 \times R_{ext} \times C_{ext}$

$t_w = 0.45 \times 180k\Omega \times 68nF$

$t_w = 5.51ms$

 Copyright: Not to be reproduced without permission

Section 4

1. In digital electronics a counter is a circuit or device that can count up and/or down in binary every time it is triggered

2. It reaches the highest number in its count sequence and the next time it is triggered it jumps back to the lowest number.

   [Note: The lowest number is usually 0000 but it doesn't have to be. It is possible using the 74LS193 to design a counter that starts from other numbers such as 0011 and counts up to 1111.]

3. It reaches the lowest number in its count sequence and the next time it is triggered it jumps back to the highest number.

   [Note: Some manufacturers prefer the term *underflow* for down-counters.]

4. Least Significant Bit

5. Most Significant Bit

6. A logic-1 on the MSB represents a bigger decimal value than a logic-1 on any of the other outputs.

   [Recall that in binary the LSB in binary is worth 1, the next number along is worth 2, next number along is worth 4, the next number 8 so on and so forth doubling each time.]

7. The number of counts in the counter's count sequence.

8. 5

9. negative edge-triggered.

10. active high enabled.

11. asynchronous.

    [Note: Based on our discussion of the operation of counters, it's a reasonably safe bet to say that the clear input is asynchronous but be careful because there are some counter ICs with synchronous clear inputs.]

12. Q_0

13. Q_4

14. 16

Section 4 (continued)

15. $Mod = 2^n$ (where n is the number of outputs)

    $Mod = 2^5$

    $Mod = 32$

16. 11111 and 31

17. Positive edge-triggered because the output of the LSB always changes from logic-0 to logic-1 and logic-1 to logic-0 on the positive edge of the clock signal.

18. An up counter because reading the logic levels on the outputs starting from the left they go 000, 111, 110, 101, 100 and so on which is a count down sequence in binary.

19. 12 because there are twelve clock pulses before the count sequence repeats.

20. $Mod = 2^n$

    $Mod = 2^{10}$

    $Mod = 1024$

21. Because the counter may have been designed to have a modulus less than 1,024 but all ten bits are still needed to make the highest number that it does count up to.

    [For example, suppose the counter only had a modulus of 1000. This means that it would count from 0 to 999 but 999 in binary is 1111100111 which requires ten bits to make.]

22. A counter made up of two or more counter ICs the outputs of which combine to make a single binary number.

23. To make counters that can count to bigger numbers than the ICs on their own can.

24. 10

25. up only.

26. positive edge-triggered.

27. active high enabled.

28. To make cascading the IC simple.

29. 4

 Copyright: Not to be reproduced without permission

Section 4 (continued)

30. both up and down.

31. positive edge-triggered.

32. To provide an input for enter a number when presetting the output.

Section 5

1. The bits that make up a data word are transmitted one at a time with serial data transfer and all at the same time with parallel data transfer.

2. It's faster.

3. Need only one wire (and common) for the data transmission.

4. D-type flip flops.

5. Frequency dividers
   Numerical multipliers
   Numerical dividers
   Delay circuits (eg for serial systems with different clock speeds)

6. negative edge-triggered.

7. 8

8. Q_4

9. Serial to parallel conversion.

10. positive edge-triggered.

11. 8

12. 11000100

13. In microprocessor-based systems
    Inside microprocessors themselves

14. Data latch

15. 1

16. 00001100

17. Parallel to serial conversion

18. active low enabled.

19. positive edge-triggered.

20. serial to parallel converter.

21. 8 bit words.

 Copyright: Not to be reproduced without permission

Section 6

1. The encoder outputs a binary number equivalent to the decimal weighting of the asserted input.

2. The decoder asserts the output with a decimal weighting is equivalent to the binary number.

3. An encoder

4. A decoder

5. An encoder

6. Both

7. Interrupt handling (in microprocessor-based systems)
   Keypad encoding

8. Address decoding (in microprocessor-based systems)
   Light chaser
   Display multiplexing

9. a decoder. [By the way, there's no such a thing as a priority decoder.]

10. active high.

11. active high.

12. active low.

13. Because the gate is logic-0 which, for this device, means that it is disabled and so none of the outputs are asserted.

14. The A and B inputs have no effect on the output when the gate is a logic-0 for this device so the logic level is irrelevant.

15. The decimal equivalent of the binary number on the input is the same as the number of the output asserted. [For example, when A and B are logic-0 output Y_0 is asserted. And, as we know, the decimal equivalent of the binary number 00 is 0.]

16. No more than one output is asserted at a time

17. a priority encoder.

18. active low.

19. active high.

Section 6 (continued)

20. Because priority encoders only react to the highest weighted input that is asserted.

21. The number that appears on the outputs is the binary equivalent of the decimal weighting of the highest asserted input.

22. The device outputs the number 000

23. All of the above

24. 4

25. 16 However, some encoder ICs dispense with input-0 because it's not needed. That being the case, 15 can also be a correct answer.

26. 3

27. 8

28. a 1-of-8 decoder with active low outputs.

29. 3

30. *G1* must be logic-1 and *G2A* & *G2B* must be logic-0

31. 21ns

32. an 8-line-to-3-line encoder with active low inputs.

33. 1

34. active low.

35. 8

36. active low.

 Copyright: Not to be reproduced without permission

Section 7

1. Time-division multiplexing involves several digital devices sharing a transmission medium (like a copper or fibre optic cable) by each device taking turns at having unshared use of the medium for short periods of time.

2. A multiplexer is a device that takes digital information from just one of several inputs and copies it to a shared output

3. Mux

4. A demultiplexer is a device that takes digital information from one shared input and copies it to just one of several outputs.

5. Demux

6. 4-input mux.

7. I_1 to I_3.

8. S_0 and S_1.

9. The data on the output is the same as the data on the data input that has the decimal number equivalent to the binary number on the S_0 and S_1 inputs.

   [For example, when S_0 and S_1 are both logic-0, the decimal equivalent of 00 is 0. So, input I_0 gets copied to the output.]

10. It doesn't matter what logic level is on the crossed input for the other input conditions shown because it won't affect the logic level on the output.

11.

S_1	S_0	Z
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

12. a decoder.

13. 4-output demux.

14. the enable.

Section 7 (continued)

15. both I_0 and I_1.

16. The input data is copied to the output that has the decimal number equivalent to the binary number on the I_0 and I_1 inputs.

    [For example, when I_0 and I_1 are both logic-0, the decimal equivalent of 00 is 0. So, the input data gets copied to output O_0.]

17. 2

18. 1 (*Y*)

19. 2 (*A* and *B*)

20. 4 (*C0*, *C1*, *C2* and *C3*)

21. A gate or enable

22. active low.

23. 1

24. 16

25. When either *G1* or *G2* is a logic-1.

26. 4 (*A*, *B*, *C* and *D*)

27. 2 (*G1* and *G2*)

28. both *G1* and *G2* for a logic-0 and either *G1* or *G2* for a logic-1.

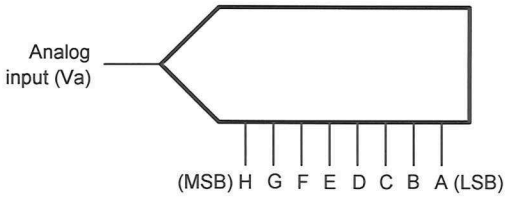
    [Note: If you're not sure about this, consider the device's function table. It shows that both *G1* and *G2* must be a logic-0 for a selected output to be a logic-0. However, only either *G1* or *G2* need be logic-1 for all outputs to become a logic-1 including the selected output.]

 Copyright: Not to be reproduced without permission

Section 8

1. Analog to digital conversion is the process of taking a sampled analog voltage or current and converting it to a binary number that is proportional to the size of the analog signal.

2.



3. The minimum and maximum analog input voltages that the ADC can convert.

4. To prevent the voltage from changing which would upset the conversion.

5. It tells you a few things actually so take your pick from the following:

   - It has more bits on its output
   - There are more steps between the minimum and maximum values in the ADC's range that can produce a unique number
   - It can distinguish between smaller portions of the same input
   - Many more of its samples will be closer to the centre value that they're compared to

6. The time it takes for the device to sample the input and convert it to a digital number.

7. Sampling rate

8. Analog inputs that are AC voltages/currents and have a high frequency.

9. 24kSa/s (or 24kSPS or 24kHz). This is always twice the frequency of the sinewave.

10. 11.2kSa/s (or 11.2kSPS or 11.2kHz). This always twice the highest frequency in a complex analog signal.

11. Look it up on the net - it's a very famous theory so you shouldn't have too much trouble finding the person's name.

12. The flash converter

13. Digitising analog video signals
    Digitising input signals to digital oscilloscopes

Section 8 (continued)

14. The majority of ADC chips are successive approximation converters.

15. The dual slope converter.

16. For its accuracy due to noise immunity.

17. Quantisation error

18. Quantisation noise

19. The classic hiss of white noise.

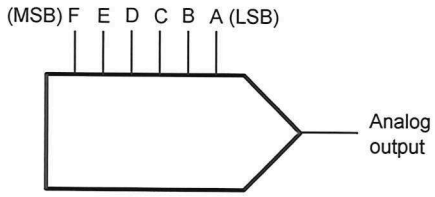
20. $Resolution = \frac{1}{2^n - 1} \times 100$

    $Resolution = \frac{1}{2^{12} - 1} \times 100$

    $Resolution = 0.0244\%$

21. Digital to analog conversion is the process of taking a binary number and converting it to a voltage or current that is (usually) proportional to the number.

22.



23. The size of the binary number that it can convert to analog.

24. How long it takes for the DAC's output voltage to change and settle in response to a change of input.

25. How much the output deviates from a straight line for a digital ramp output.

26. $Resolution = \frac{1}{2^n - 1} \times 100$

    $Resolution = \frac{1}{2^{10} - 1} \times 100$

    $Resolution = 0.098\%$